Contents

Preface ................................................................. 1
  About This Manual .............................................. 1
  Overview .......................................................... 1
  Intended Audience .............................................. 2
  Text Conventions ................................................ 2
  Other Documentation .......................................... 3

Chapter 1 Key Concepts ........................................... 1
  Connection Types ................................................. 1
  About WS-Trust STS ............................................. 2
  Connection-Based Policy ....................................... 2
    IdP Configuration ............................................. 2
    SP Configuration ............................................. 2
  Token Processors and Generators ............................. 3
    Bundled Token Plug-ins ..................................... 3
    Commercial Token Plug-ins ................................. 3
  Client SDKs ....................................................... 3
  SSO Integration Kits and Adapters ......................... 4
    Bundled Adapters ............................................. 4
    Commercial Adapters ....................................... 4
    Software Development Kit .................................. 4
  Identity Mapping ................................................ 5
  Account Linking ................................................ 5
    Linking Permission and “Defederation” ................... 5
  SP Affiliations ................................................. 5
  Account Mapping ................................................ 5
  About Attributes ............................................... 6
  Attribute Contracts .......................................... 7
  Name Formats ................................................... 7
## Contents

- Adapter Contracts ......................................................... 8
- Extended Adapter Contract .............................................. 8
- STS Token Contracts ....................................................... 8
- Extended Token Generator Contract .................................... 9
- Data Stores ..................................................................... 9
- Attribute Masking ............................................................ 9
- Certificates, SSL, and XML Encryption .................................. 10
- Digital Signatures ............................................................. 10
- Message Signing .............................................................. 10
- Signature Validation .......................................................... 11
- Certificate Validation ........................................................ 11
- Digital Signing Policy Coordination ...................................... 12
- Secure Sockets Layer ........................................................ 13
- SAML SSL/TLS Scenarios .................................................. 13
- Authentication ................................................................. 13
- Verifying Trusted Certificates .............................................. 14
- XML Encryption ............................................................... 14
- About PingFederate Express .............................................. 14
- Using Auto-Connect .......................................................... 15
- Providing Metadata ........................................................... 15
- Runtime Processing ........................................................... 15
- Auto-Connect Security Model ............................................. 17
- User Provisioning .............................................................. 17
- SaaS Provisioning .............................................................. 18
- Express Provisioning .......................................................... 18
- Federation Planning Checklist ............................................ 19
- Configuration Data Exchange .............................................. 21
- IdP to SP ...................................................................... 21
- SP to IdP ...................................................................... 21
- Mutual Settings Between Parties ........................................ 22
- Chapter 2 System Administration .......................................... 23
- Starting and Stopping PingFederate ....................................... 24
- Log File Generation ............................................................. 24
- Administrator Audit Logging .............................................. 25
- Runtime Transaction Logging ............................................. 26
- Transaction Logging Modes ............................................... 27
- Exporting Metadata ............................................................ 27
- Defining Metadata Attribute Contracts ................................ 29
- Choosing a Metadata Signing Key ....................................... 30
- XML Encryption Certificates .............................................. 30
- Completing the Export ....................................................... 30
- Signing XML Files ............................................................. 31
- Using the Configuration Archive Utility ................................. 32
- Using the Archive Import Screen ....................................... 33
- Account Management ........................................................ 34
Table of Contents

Chapter 4  Security Management ................................................. 87
  Trusted Certificate Authorities ........................................... 87
  SSL Server Certificates ...................................................... 88
  SSL Client Keys & Certificates ............................................ 91
  Digital Signing and Decryption Keys & Certificates ................. 93
  Application Authentication .................................................. 95
  Certificate Revocation Checking ......................................... 97
  SSL Client Keys & Certificates ............................................ 91
  SSL Server Certificates ...................................................... 88
  Trusted Certificate Authorities ........................................... 87
  Application Authentication .................................................. 95
  Certificate Revocation Checking ......................................... 97

Chapter 5  Identity Provider SSO Configuration .......................... 101
  Application Integration Settings .......................................... 102
  Configuring IdP Adapters .................................................... 102
    Selecting an IdP Adapter Type .......................................... 103
    Configuring an IdP Adapter .............................................. 104
    Invoking Adapter Actions ............................................... 104
    Extending an Adapter Contract ........................................ 105
    Setting Pseudonym Values and Masking ................................ 106
    Selecting an Authentication Context ................................... 106
    Editing and Saving Adapter Instances ................................ 107
  Configuring a Default URL and Error Message ........................ 107
  Viewing Application Endpoints ............................................ 108
  Viewing Protocol Endpoints ............................................... 108
  Managing SP Connections ................................................... 110
    Accessing Connections .................................................... 110
    Using the Main Menu ...................................................... 110
    Using the Manage Connections Screen ................................ 111
    Choosing a Connection Template ....................................... 114
    Choosing a Connection Type ............................................ 114
    Choosing Connection Options .......................................... 115
  Importing Metadata ........................................................ 116
    Importing a Verification Certificate .................................. 116
    Viewing the Metadata Summary ........................................ 117
  General Information ........................................................ 117
  Configuring Browser-Based SSO ........................................... 119
    Configuration Steps ....................................................... 120
    Choosing Profiles (SAML 2.0) .......................................... 121
    Setting an Assertion Lifetime .......................................... 124
    Assertion Creation ........................................................ 124
    Configuring Protocol Settings ......................................... 147
    Editing and Saving Browser SSO Settings ............................. 156
  Configuring the Attribute Query Profile ............................... 157
    Defining Retrievable Attributes ....................................... 157
    Choosing a Data Store .................................................... 158
    Configuring Data Store Lookup ........................................ 158
    Attribute Mapping Fulfillment ......................................... 159
    Specifying Security Policy .............................................. 160
    Editing and Saving Attribute Query Configurations ................ 160
# Chapter 6 Service Provider SSO Configuration

- Configuring Credentials .................................................. 160
- Configuring Back-Channel Authentication .............................. 161
- Configuring Digital Signature Settings ................................. 164
- Configuring Signature Verification Settings ........................... 165
- Selecting an Encryption Certificate (SAML) ......................... 168
- Selecting a Decryption Key (SAML) .................................. 169
- Editing and Saving Credential Configurations ....................... 170
- Configuring SaaS Provisioning ........................................... 170
- Defining a Provisioning Target .......................................... 171
- Managing SaaS Channels .................................................. 172
- Specifying Channel Information ........................................ 173
- Identifying the Source Data Store ...................................... 174
- Modifying Source Settings ................................................ 174
- Specifying a Source Location ............................................. 177
- Mapping Attributes .......................................................... 178
- Channel Activation and Summary ...................................... 181
- Connection Activation and Summary ................................... 182
- Defining SP Affiliations .................................................... 183
- Using the Manage Affiliations Screen ................................ 183
- Importing Affiliation Metadata .......................................... 184
- Entering Affiliation Information ......................................... 184
- Managing Affiliation Membership ...................................... 185
- Using the Express Connection Template ............................... 186
- Finishing an Express Connection ....................................... 187
- Configuring SP Auto-Connect ............................................ 189
- Initial Setup ..................................................................... 189
- Setting an Assertion Lifetime ............................................. 190
- Choosing a Signing Certificate .......................................... 190
- Configuring Assertion Creation ......................................... 191
- Auto-Connect Activation and Summary ............................... 191
- Specifying Allowed SP Domains ........................................ 192

## Chapter 6 Service Provider SSO Configuration

- Application Integration Settings ........................................ 193
- Configuring SP Adapters .................................................... 194
- Creating an Adapter Instance ............................................. 195
- Configuring an Adapter Instance ........................................ 197
- Invoking Adapter Actions .................................................. 197
- Extending an Adapter Contract .......................................... 197
- Editing and Saving SP Adapter Instances ............................. 198
- Mapping URLs to Adapter Instances ................................... 198
- Configuring Default URLs ................................................ 200
- Setting Target Validation .................................................. 201
- Viewing Application Endpoints ......................................... 202
- Federation Settings ........................................................... 203
- Attribute Requester Mapping ............................................. 203
- Viewing Protocol Endpoints .............................................. 205
Contents

Managing IdP Connections ................................................. 206
Accessing Connections .................................................. 207
Using the Main Menu ..................................................... 207
Using the Manage Connections Screen ................................ 208
Choosing a Connection Type ............................................. 210
Choosing Connection Options ......................................... 210
Importing Metadata ....................................................... 211
Importing a Verification Certificate .................................... 211
Viewing the Metadata Summary ......................................... 212
General Information ........................................................ 212
Configuring Browser-Based SSO ........................................ 214
Configuration Steps ....................................................... 214
Choosing SAML Profiles ................................................. 216
User-Session Creation .................................................... 218
Configuring Protocol Settings .......................................... 234
Editing and Saving Browser SSO Settings ............................. 244
User Provisioning ............................................................ 244
Choosing an Event Trigger ............................................... 245
Selecting Attribute Sources (SAML 2.0) .............................. 245
Identifying the User Repository ........................................ 246
Specifying an LDAP User-Record Location ......................... 246
Specifying an LDAP Filter .............................................. 247
Identifying Provisioning Attributes for LDAP ...................... 248
Specifying a Database User-Record Location ....................... 248
Specifying a Unique-ID Database Column ............................ 249
Mapping Attributes to User Accounts .................................. 250
Error Handling .............................................................. 251
Using the Provisioning Summary Screen ............................... 252
Configuring the Attribute Query Profile ............................... 252
Setting the Attribute Authority Service URL ......................... 252
Mapping Attribute Names ............................................... 253
Specifying Security Policy .............................................. 254
Editing and Saving Attribute Query Configurations .................. 254
Configuring Credentials .................................................. 254
Configuring Back-Channel Authentication ............................ 255
Configuring Digital Signature Settings ............................... 258
Configuring Signature Verification Settings ......................... 258
Selecting an Encryption Certificate .................................... 262
Selecting a Decryption Key ............................................. 264
Editing and Saving Credential Configurations ....................... 264
Connection Activation and Summary .................................... 265
Configuring IdP Auto-Connect ........................................... 265
Configuring the Initial Setup ............................................ 266
Choosing a Signing Certificate ......................................... 266
Configuring User-Session Creation ..................................... 266
Chapter 7  WS-Trust STS Configuration .................................................. 269
   Server Settings ................................................................. 269
   Enabling the WS-Trust STS ............................................... 269
   Configuring STS Authentication ......................................... 271
      Selecting Authentication Methods .................................. 271
      Configuring Basic Authentication .................................. 271
      Configuring Mutual SSL Authentication ............................ 272
      Using the WS-Trust STS Settings Summary Screen ........... 275
   IdP Configuration for STS .................................................. 275
      Configuring Token Processors ........................................ 276
         Selecting a Token Processor Type ................................ 276
         Configuring a SAML Token Processor Instance .............. 277
         Extending a Processor Contract .................................. 279
         Setting Attribute Masking ........................................... 279
         Editing and Saving Processor Instances ....................... 279
      Managing STS Request Parameters ................................ 280
         Creating a Request Contract ....................................... 280
      Configuring SP Connections for STS ............................... 281
         Configuring Protocol Settings .................................... 282
         Setting a Token Lifetime ............................................ 283
         Configuring Token Creation ....................................... 284
   SP Configuration for STS .................................................. 300
      Configuring Token Generators ....................................... 300
         Selecting a Token Generator Type ............................... 301
         Configuring a Token Generator Instance ....................... 301
         Extending a Generator Contract .................................. 302
         Editing and Saving Generator Instances ....................... 303
      Configuring IdP Connections for STS ............................. 303
         Configuring Protocol Settings .................................... 304
         Configuring Token Generation .................................... 304
   Appendix A  OpenToken Adapter Configuration .......................... 319
      Configuring the IdP OpenToken Adapter ......................... 320
      Configuring the SP OpenToken Adapter ......................... 323
   Appendix B  LDAP Adapter Configuration ................................ 327
      Configuring the IdP LDAP Adapter ................................ 329
      Configuring the SP LDAP Adapter ................................ 332
   Appendix C  Application Endpoints ...................................... 335
      IdP Endpoints ............................................................ 335
      SP Endpoints ............................................................ 338
      Maintenance Endpoint ................................................... 343
   Appendix D  Web Service Interfaces ...................................... 345
      Connection Management Service ................................... 345
      Exporting a Connection ............................................... 346
## Contents

- Exporting Manually ................................................................. 346
- Using the Connection Service .................................................... 346
- Importing Connections ............................................................ 347
  - Code Sample ........................................................................ 347
- Deleting Connections ............................................................... 348
  - Code Sample ........................................................................ 348
- Cluster Configuration Replication ................................................. 348
  - Code Sample ........................................................................ 348
- Validation Disclaimer ................................................................. 349
- SSO Directory Service ............................................................... 349
  - Code Example ...................................................................... 350
- SOAP Request and Response Example ......................................... 351

### Appendix E
- Using Attribute Mapping Expressions ............................................ 353
  - About OGNL ...................................................................... 353
  - Enabling/Disabling Expressions .............................................. 354
  - Using the OGNL Edit Screen .................................................. 355

### Appendix F
- Troubleshooting ....................................................................... 357
  - Data Stores ....................................................................... 357
  - Installation ......................................................................... 358
  - Protocols ........................................................................... 358
  - Server ............................................................................... 358
- Glossary .................................................................................. 361
- List of Acronyms ..................................................................... 367
Preface

About This Manual

This manual provides information about using Ping Identity’s PingFederate to deploy a secure Internet single sign-on (SSO) solution based on the latest security and e-business standards.

Overview

The manual consists of:

- **Chapter 1, “Key Concepts”** — A discussion of central concepts needed to understand Internet SSO, the WS-Trust Security Token Service (STS), and PingFederate deployment and administration.
- **Chapter 2, “System Administration”** — Information about maintaining the PingFederate server and deployment, using log files, managing users, and handling other administrative functions.
- **Chapter 3, “System Settings”** — How to configure your local PingFederate server settings.
- **Chapter 4, “Security Management”** — Information about importing, exporting, and maintaining certificates and keys in PingFederate.
- **Chapter 5, “Identity Provider SSO Configuration”** — How to configure PingFederate to act as an Identity Provider (IdP) and establish connections to Service Providers.
- **Chapter 6, “Service Provider SSO Configuration”** — How to configure PingFederate to act as a Service Provider (SP) and establish connections to Identity Providers.
- **Chapter 7, “WS-Trust STS Configuration”** — How to configure PingFederate to act as a Security Token Service for Web Service Clients and Providers in either an IdP or SP environment.
- **Appendix A, “OpenToken Adapter Configuration”** — How to configure PingFederate to use the packaged OpenToken Adapter for interfacing with your Web applications.
Preface

- Appendix B, “LDAP Adapter Configuration” — How to configure PingFederate to use the packaged LDAP Authentication Adapter.
- Appendix D, “Web Service Interfaces” — Information developers can use to automate connection-configuration management and runtime SSO partner discovery.
- Appendix E, “Using Attribute Mapping Expressions” — How to enable and use expressions in conjunction with mapping attributes.
- Appendix F, “Troubleshooting” — Solutions for difficulties that may be encountered.
- Glossary — Definitions of terms used in the manual and in identity federation parlance.
- List of Acronyms

Intended Audience

This manual is intended for security and network administrators and other IT professionals responsible for identity management among business entities, both internal and external.

Note: The information in this manual is presented from the viewpoint of an administrative user with full permissions (see “Account Management” on page 34).

Text Conventions

This document uses the text conventions identified below.

Table 1: Text Conventions

<table>
<thead>
<tr>
<th>Convention</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fixed Width</td>
<td>Indicates text that must be typed exactly as shown in the instructions. Also used to represent program code, file names, and directory paths.</td>
</tr>
<tr>
<td>Blue text</td>
<td>Indicates hypertext links.</td>
</tr>
<tr>
<td>Italic</td>
<td>Used for emphasis and document titles.</td>
</tr>
<tr>
<td>[text]</td>
<td>Used for procedures where only one step is required.</td>
</tr>
<tr>
<td>Sans serif</td>
<td>Identifies descriptive text on a user-interface screen. Example: “Print Document dialog”</td>
</tr>
<tr>
<td>Sans serif bold</td>
<td>Identifies menu items, navigational links, or buttons. For example: Click Save.</td>
</tr>
</tbody>
</table>
Other Documentation

Unless otherwise noted, the documents listed below are located in your PingFederate installation's pingfederate/docs directory.

Tip: PingFederate provides context-sensitive online Help. Click Help in the upper-right portion of the administrative console for immediate, relevant guidance and links to related information.

**Getting Started** – Provides an introduction to secure Internet SSO and PingFederate, including background information about federated identity management and standards, product installation instructions, and a primer on using the PingFederate administrative console.

**Quick-Start Guide** – Provides instructions for deploying a preconfigured PingFederate server to run with example Web applications. Ping Identity recommends that you follow this Guide as a first step to establishing a simple identity federation between two Web applications and to familiarize yourself with PingFederate. The Guide is located in the quickstart/docs directory.

**Integration Overview** – A high-level description of options available for integrating identity-management systems and applications with PingFederate.

**Server Clustering Guide** – Describes how to deploy PingFederate in a cluster to increase throughput and availability.

**SDK Developer's Guide** – Provides technical guidance for using the Java Software Developer Kit for PingFederate version 4 and higher. This Guide is located in the pingfederate/sdk directory.

**Web Resources** – Ping Identity continually updates its Web site with general and technical information in the form of White Papers, FAQs, Tech Notes, and other resources—[www.pingidentity.com](http://www.pingidentity.com).

PingFederate documents may include hypertext links to Web sites that provide installation instructions, file downloads, and reference documentation. These links were tested prior to publication, but they may not remain current throughout the life of these documents. Please contact Ping Identity Support (support@pingidentity.com) if you encounter a problem.
Preface
Key Concepts

This chapter provides background information and preparation to help you understand and use PingFederate:

- “Connection Types” below
- “About WS-Trust STS” on page 2
- “SSO Integration Kits and Adapters” on page 4
- “Identity Mapping” on page 5
- “About Attributes” on page 6
- “Certificates, SSL, and XML Encryption” on page 10
- “About PingFederate Express” on page 14
- “Using Auto-Connect” on page 15
- “User Provisioning” on page 17
- “Federation Planning Checklist” on page 19

**Tip:** For an introduction to secure Internet single sign-on (SSO), federated identity management, and PingFederate product features, see the *Getting Started* manual in the `pingfederate/docs` directory.

Connection Types

PingFederate features an integrated administrative console for configuring two kinds of connections to identity-federation partners:

- **Browser-based SSO** – Also called Browser SSO in the administrative console, this term is often used to refer to standards-based secure Internet SSO, which generally depends on a user's browser to transport identity assertions and other messaging between partner endpoints (see the “Supported Standards” chapter in *Getting Started*).
Chapter 1
Key Concepts

- WS-Trust STS – This type of connection employs the PingFederate Security Token Service (STS), which enables Web Service Client and Provider applications to extend SSO to identity-enabled Web Services at provider sites, using another set of standards (see the next section, “About WS-Trust STS”). These standards, including WS-Trust, do not rely on the user’s browser for message transport.

The two types of connections can be configured together for the same partner or independently (see “WS-Trust STS Configuration” on page 269).

About WS-Trust STS

The PingFederate WS-Trust STS allows organizations to extend SSO identity management to Web Services. (For information about WS-Trust and the role of an STS, see “Web Services Standards” in the “Supported Standards” chapter of Getting Started.)

The WS-Trust STS can be configured for partner connections independently or in conjunction with browser-based SSO for either an IdP or an SP deployment. The STS is bundled with separate plug-ins for standard SAML (Security Assertion Markup Language) token processing and generation (see “Token Processors and Generators” on page 3).

Connection-Based Policy

For both the IdP and SP roles, PingFederate employs a partner-connection configuration, which enables the association of Web Services authentication policies with federation partners. For STS processing, these policies define configurations for handling WS-Trust requests and transferring identity information between security domains (see “Web Services Standards” in the “Supported Standards” chapter of Getting Started).

IdP Configuration

In an IdP role, you use the administrative console to configure WS-Trust request-processing policy for your SP partner including:

- The type of SAML token to create—suitable for consumption by the intended Web Service Provider (at the SP site)—in response to an “Issue” request from a Web Service Client application
- The mapping of attributes to include within the issued SAML token
- The key used to create a digital signature for the issued SAML token

SP Configuration

In an SP role, you use the administrative console to configure WS-Trust request-processing policy for your IdP partner including:

- Whether to validate the incoming SAML token only, or to validate the incoming token and also issue a local token
- The mapping of attributes to include in the locally issued token (when applicable)
- The certificate used to verify the digital signature for the incoming SAML token
- The key used to decrypt the incoming SAML token (when needed)
Token Processors and Generators

PingFederate provides support for a variety of security-token formats, through token processors and generators that plug into the PingFederate server. These plug-ins deploy similarly to browser-based SSO adapters (see “SSO Integration Kits and Adapters” on page 4).

For an IdP, token processors provide a mechanism through which PingFederate can validate an incoming token and map attributes to be included in the issued SAML token.

For an SP, token generators provide a mechanism through which PingFederate can generate a local token based upon the incoming SAML token and map attributes to be included in that token.

Only SAML 1.1 or 2.0 tokens are generated by PingFederate configured as an IdP for sending across trust boundaries to a federated SP partner. Likewise, only SAML tokens are accepted by PingFederate configured as an SP. Token plug-ins allow a modular approach for validating and producing the various token types used by different applications or systems within a conceptual trust domain. PingFederate provides bundled and separately available token plug-ins.

Token processors (for an IdP) and generators (for an SP) are configured globally in the administrative console; and at runtime, only a single instance of each token type can be active at one time.

Bundled Token Plug-ins

PingFederate is installed with token processors for an IdP configuration that accept and validate SAML 1.1 and 2.0 tokens (SAML tokens are issued on the IdP side via built-in browser-based SSO capabilities).

For an SP configuration, token generators are provided for issuing local SAML 1.1 or 2.0 tokens (incoming SAML tokens are validated, once again, by using built-in capabilities).

Commercial Token Plug-ins

Separately available token plug-ins include:

- **Username** – Accepts and validates Web Services Security (WSSE) Username tokens against a password or an LDAP v3-compliant directory
- **X.509** – Accepts and validates WSSE X.509 tokens against the PingFederate trust store
- **Kerberos** – Accepts and validates Kerberos binary tokens
- **CA SiteMinder** – Accepts and validates SMSESSION binary tokens
- **Oracle Access Manager (formerly COREid)** – Accepts and validates OBSSO binary tokens
- **OpenToken** – Accepts and validates an OpenToken at an IdP PingFederate and issues an OpenToken at the PingFederate SP

Ping Identity regularly develops token plug-ins to work with various authentication systems and leading identity management systems. Also known as Token Translators, available plug-ins are posted on our Web site (see the PingFederate Overview page).

Client SDKs

Ping Identity provides a set of Java and .NET WSE3 client Software Development Kits (SDKs) for enabling Web Service applications (client or provider) to interact with the PingFederate STS.
Chapter 1

Key Concepts

The SDKs provide interfaces that create the WS-Trust Request Security Token (RST) and Request Security Token Response (RSTR) messaging to interact with the PingFederate STS endpoints. Using the SDK libraries, applications are not responsible for forming these WS-Trust messages, and instead interact only with the tokens themselves.

SSO Integration Kits and Adapters

As a stand-alone server, PingFederate must be programmatically integrated with end-user applications and identity management (IdM) systems to complete the “first- and last-mile” implementation of a federated identity network for browser-based SSO.

For an IdP (the first mile), this integration process involves providing a mechanism through which PingFederate can look up a user's current authenticated session data (for example, a cookie) or authenticate a user without such a session. For an SP, the last mile involves enabling PingFederate to supply information needed by the target application to set a valid session cookie or other application-specific security context for the user.

To enable both sides of this integration, PingFederate provides bundled and commercial integration kits, which include adapters that plug into the PingFederate server and agent toolkits that interface with local IdM systems or applications.

PingFederate includes a robust software development kit (SDK), which software developers can use to write their own adapters for specific systems. Adapters can be written to retrieve attributes from custom data stores, connect to application- or IdM-specific user authentication systems, or provide complex attribute transformations or processing.

Bundled Adapters

PingFederate packages two adapters:

- An OpenToken Adapter, which provides a generic interface for integrating with various applications, including Java- and .NET-based applications (see “OpenToken Adapter Configuration” on page 319)
- An LDAP Authentication Service, which interfaces with LDAP v3-compliant directories (see “LDAP Adapter Configuration” on page 327)

Commercial Adapters

Ping Identity regularly develops integration kits, including adapters, to work with applications and leading identity management systems. Available kits are posted on our Web site at www.pingidentity.com.

Software Development Kit

The PingFederate SDK provides a flexible means of creating custom integration kits to integrate federated identity management into your system environment. See the PingFederate SDK Developer's Guide in the /sdk directory for more information.
Identity Mapping

Identity mapping is at the core of identity federation. One of the primary goals of SAML is to provide a way for an identity provider (IdP) to send a secure token (the assertion) containing user-identity information that a service provider (SP) can translate, or map, to local user stores. (For more information about SAML, see the “Supported Standards” chapter in Getting Started.)

For browser-based SSO, PingFederate enables two modes of identity mapping between domains:

- Account Linking
- Account Mapping

For WS-Trust STS, account mapping is used.

Account Linking

Under the standards, account linking can be used for browser-based SSO in cases where each domain maintains separate accounts for the same user. Account linking uses the SAML assertion to create a persistent association between these distinct user accounts. The account link, or name identifier, may be either a unique attribute, such as an email address, or a pseudonym generated by the IdP to uniquely identify individual users. Pseudonyms can be used when privacy is a concern; they cannot easily be traced back to a user's identity at the partner site.

During the user's first SSO request, the SP prompts for local credentials, which enables the SP to link the name identifier contained within the assertion—either an open attribute or a pseudonym—with the user's local account. Subsequent SSO events will not prompt the user to authenticate with the SP, since the SP federation server keeps a table associating remote users' name identifiers with local user accounts. The SP associates the link to the user's corresponding local account and provides access to the account without separate authentication.

Optionally, additional attributes may be sent with the name identifier. When a pseudonym is used as the account link, however, care must be taken to send only general attributes (a user's organizational role or department, for example) that will not compromise privacy.

Linking Permission and “Defederation”

The SAML specification also allows the SP application to build in user verification and approval of account linking and provides a means for the user to permanently cancel the linking, known as defederation (see “/sp/defederate.ping” on page 341). A user who has defederated may later elect to reassociate with a local user account.

SP Affiliations

Under the SAML 2.0 specifications, an IdP can configure PingFederate to enable a group of SPs—an SP affiliation—to share the same persistent name identifier (see “Defining SP Affiliations” on page 183). This capability facilitates the use case in which a number of business partners have an existing relationship and sharing a single name identifier among all parties reduces the federation integration effort.

Account Mapping

Account mapping (also called “attribute mapping”) enables an SP to use PingFederate to perform a user lookup and map a user's identity dynamically based on one or more attributes received in the assertion. The attributes used to look up the user are always...
Chapter 1

Key Concepts

“exposed”; that is, they are known to both the IdP and SP. An email address, for example, is a commonly used identifying attribute.

Account mapping can be used to achieve one-to-one mapping (individual user accounts exist on both sides of federated connection) or many-to-few (IdP users without accounts at destination sites may be mapped to guest accounts or to a role-based general account).

For browser-based SSO, transient identifiers provide an additional level of privacy—virtual anonymity—by generating a different opaque ID each time the user initiates SSO. Transient IDs are often used in conjunction with federation role mapping, whereby the user is mapped to a guest account or to a role-based account based on the user’s association with the IdP organization rather than personal attributes.

As with pseudonyms, additional attributes may be sent with the transient identifier. Again, care should be taken to preserve privacy.

Account mapping is commonly implemented in B-to-B or B-to-E use cases where it might be appropriate for the administrator to create a user lookup on behalf of the user.

About Attributes

Federation transactions require, at a minimum, the transmission of a unique piece of information (such as an email address) that identifies the user for identity mapping between security domains.

In addition to attributes used for identity mapping, the IdP can pass other user attributes in an assertion (including SAML tokens for Web Services). This supplemental information can be used by the SP for several purposes. For example, attributes may be used to map and authorize the user into a specific role, with associated site permissions. In other cases, attributes may be used to customize the end application display for a more robust user experience.

The SP also has the option of incorporating additional attributes prior to creating a session for the target application. This is commonly done where the SP also maintains an account for the user and wants to pass additional information for profiling or access-policy purposes.

Attributes must be carefully managed between IdPs and SPs. PingFederate facilitates the process by providing configuration steps that enable administrators to:

- Define and enforce attribute contracts for each partner connection.
- Define and retrieve attributes from the adapter or STS token processor to populate an attribute contract directly or use these attributes to look up additional attributes in IdP data stores.
- Define and enforce a set of required attributes needed by SP adapters or STS token generators to interface local systems or applications (see “Adapter Contracts” on page 8).
- Set up connections to local data stores (see “Data Stores” on page 9).
- Configure specific attribute sources and lookups—based on the data stores—and map attributes into IdP assertions or into SP adapters or token generators used to interface target applications (see “SSO Integration Kits and Adapters” on page 4 or “Token Processors and Generators” on page 3).
- Selectively mask attribute values recorded in transaction logs (see “Attribute Masking” on page 9).
Attribute Contracts

An attribute contract represents an agreement between an SP and an IdP about user attributes sent in a SAML assertion, either for browser-based SSO or WS-Trust STS. The contract is a list of case-sensitive attribute names. IdPs and SPs must configure attribute contracts to match.

Tip: When privacy is required for sensitive attributes, you can configure PingFederate to mask their values in log files (see “Attribute Masking” on page 9).

For an IdP, the attribute contract defines which attributes PingFederate sends in an assertion. While this contract is fixed for all users authenticating to the SP partner, the values used to fulfill the contract may differ from one user to the next. The attribute contract may be fulfilled by relying on a combination of different data sources:

- The IdP adapter or STS token processor
- An IdP attribute source, which identifies the location of individual attributes in a data store
- Static text values for some attributes, or text values combined with variables
- Expressions (see “Using Attribute Mapping Expressions” on page 353)

For an SP, the attribute contract defines the attributes PingFederate expects in a SAML assertion. PingFederate can be configured to pass these attributes to the SP adapter or, for Web Services, to the SP token generator (see “Configuring SP Adapters” on page 194 or “Configuring Token Generators” on page 300). You can also use attributes to look up additional attributes in local data stores, which may be needed to start a user session or create a local security token for Web Services (see “Adapter Contracts” below or “STS Token Contracts” on page 8).

The attribute contract must contain the attribute `SAML_SUBJECT`, the primary information used to identify the user, unless you are using account linking for browser-based SSO. This attribute is automatically included when creating a new contract.

Note: You create attribute contracts on a per-connection basis. For example, if an SP has deployed two session-creation adapters for two separate applications, a single attribute contract can be created for the IdP connection partner. This single contract would be constructed to supply all the attributes needed by both SP adapters.

Name Formats

By agreement with an SP partner, an IdP may specify a format (email, for example) associated with the `SAML_SUBJECT`. The SP may require this information to facilitate handling of the format.

For SAML 2.0, the partner agreement may also include a requirement for the IdP to provide format specifications associated with other attributes.
Chapter 1

Key Concepts

For browser-based SSO connections, PingFederate provides a means for an IdP administrator to select from among standard subject and/or attribute formats, depending on the relevant SAML specifications.

**Note:** The designation of formats is not applicable to SP administrators. The information is simply available in the incoming assertion to an SP application that might need it for particular processing requirements.

For the WS-Trust IdP configuration, attribute-name formats cannot be specified. If needed, however, an administrator can use a special variable in the attribute contract to set the subject-name format (see “Defining an Attribute Contract” on page 285). (The same variable is also available for browser-based SSO attribute contracts, but the feature is deprecated.)

**Adapter Contracts**

An adapter contract represents an agreement between the PingFederate server and an external application. In concert with the attribute contract between partners, adapter contracts specify the transfer of attributes. Adapter contracts consist of a list of case-sensitive attribute names.

On the IdP side of a federation, adapter attributes are supplied to PingFederate by an IdP adapter (see “SSO Integration Kits and Adapters” on page 4 and “Configuring IdP Adapters” on page 102).

On the SP side, adapter contract attributes are those required by an adapter to start a session with an application. At least one adapter type is needed for each security domain. Then an adapter instance must be configured for each target application. (See “Configuring SP Adapters” on page 194.)

Adapter contracts on the SP side are fulfilled using attributes from the attribute contract, possibly enhanced through other attributes looked up from local data stores. For example, if several target applications are controlled by the same security context (for example, SiteMinder) and can receive the same set of attributes to start a session for the user, you would deploy an adapter type and configure an adapter instance for each protected application (see “Configuring Adapter Mapping and User Lookup” on page 222).

**Extended Adapter Contract**

Adapter contracts are created when an adapter type is deployed with PingFederate. When developed, these adapters are “hard-wired” to look up or set a specific set of attributes. After deployment, your attribute requirements may change. To streamline adjustment of adapter contracts, PingFederate allows an administrator to add additional attributes to the adapter instance through the administrative console. These adjustments are called extended attribute contracts.

**STS Token Contracts**

Similar to an adapter contract for browser-based SSO, an STS token-processor or token-generator contract represents an agreement between the PingFederate server and an external application in the context of a Web Services transaction. In concert with the attribute contract between partners, token contracts specify the transfer of attributes, consisting of a list of case-sensitive attribute names.

On the IdP side of a federation, token-processor attributes are supplied to PingFederate (see “Token Processors and Generators” on page 3 and “Configuring Token Processors” on page 276).
On the SP side, token-generator contract attributes are those required by a token generator to pass identity information from the token to the Web Service client application. At least one token generator type is needed for each security domain. Then a token generator instance must be configured for each target application (see “Configuring Token Generators” on page 300. If several target applications are controlled by the same security context (for example, SiteMinder) and can receive the same set of attributes for the user, you would deploy a token generator type and configure a token generator instance for each target application (see “Mapping Token Generators” on page 306).

**Extended Token Generator Contract**

Token-generator contracts are created when a token-generator type is deployed with PingFederate. When developed, these token generators are “hard-wired” to look up or set a specific set of attributes. After deployment, your attribute requirements may change. To streamline adjustment of token-generator contracts, PingFederate allows an administrator to add additional attributes to the token-generator instance through the administrative console. These adjustments are called extended token-generator contracts.

**Data Stores**

PingFederate can be configured to use local data stores to supply attributes for either the IdP’s attribute contract, the SP’s adapter contract, or STS token contracts (see sections above). Standard data stores may include any JDBC-accessible database or an LDAP v3-compliant directory server (see “Managing Data Stores” on page 72).

Alternatively, you can use the PingFederate Custom Source SDK to create your own driver for non-JDBC/LDAP data stores—including, for example, flat files or SOAP-connected databases (see the PingFederate SDK Developer’s Guide in the pingfederate/sdk directory).

Data stores can be used across multiple connections.

**Attribute Masking**

At runtime PingFederate logs user attributes (see “Log File Generation” on page 24). To preserve user privacy, you may wish to mask the values of logged attributes.

PingFederate provides this masking capability at all points where the server logs attributes. These points include:

- Data-store lookup at either the IdP or SP site (see “Managing Data Stores” on page 72).
- Retrieval of attributes from an IdP adapter or token processor (see “Setting Pseudonym Values and Masking” on page 106 and “Setting Attribute Masking” on page 279).
- SP-server processing of incoming attributes based on the SSO attribute contract (see “Creating an Attribute Contract” on page 220).

Note that the SAML Subject ID is not masked: the SAML specifications provide for either pseudonymous account linking or transient identification to support privacy for the Subject ID (see “Account Linking” on page 5).

- SP-server processing of incoming attributes in response to an Attribute Request under XASP (see “Specifying Security Policy” on page 254).

For information about XASP, see “Attribute Query and XASP” in the “Supported Standards” chapter of Getting Started.
Chapter 1

Key Concepts

Certificates, SSL, and XML Encryption

This section describes the PingFederate security infrastructure that supports encrypted messaging, certificates, and digital signing. These functions are integrated into PingFederate’s configuration screens to provide complete control over certificate generation and authentication verification (see “Security Management” on page 87).

Digital Signatures

A digital signature is a way to verify the identity of a person or entity who originates an electronic document and ensure that the message has not been altered. Digital signatures are used in both SAML (including STS tokens) and WS-Federation electronic documents.

Handling a digital signature involves message signing, signature and certificate validation, and signing-policy coordination between connection partners.

Message Signing

Certificates contain information about the owner of the certificate along with a public key. Applying a digital signature creates and encrypts a hash from the message you are signing, using your private key.

To ensure the integrity of SAML messages or STS tokens, we recommend digital signing practices, using public/private keypairs in conjunction with X.509 certificates.

Important: Many adapter implementations, as well as other product extensions, may independently write unmasked attribute values to the PingFederate server log. These implementations are beyond the control of PingFederate. If sensitive attribute values are a concern when using such a component, a system administrator can adjust the component’s logging threshold in \texttt{log4j.xml} to prevent the recording of attributes (see “Log File Generation” on page 24).

Note: Digital signatures do not encrypt the contents of a message; SSL/TLS and/or XML encryption is used for this purpose.

The certificate should be signed by a Certificate Authority (recommended), but it can be self-signed or signed by an untrusted third party. After generating a keypair and a self-signed certificate, you can use PingFederate to create a Certificate Signing Request (CSR) and send it to a CA for signing. After the CA has generated a Certificate Signing Response, you can import it into PingFederate’s certificate management system. (The CA’s certificate must be in PingFederate’s trusted store or in the Java runtime cacerts store.)

PingFederate enables signing and validation of responses, requests, and/or the assertion message. In addition, PingFederate provides for certificate generation, import and export functionality, CSR generation, and application of digital signatures. You can create reusable global signing certificates across your federated...
connection base and import signature verification certificates for each partner (see “Digital Signing and Decryption Keys & Certificates” on page 93).

**Note:** Ping Identity recommends generating unique certificates for each connection, which limits exposure if the private key becomes compromised.

### Signature Validation

After receiving a signed message, PingFederate verifies the signature using the public key that corresponds with the private key used to sign the message or token. Verification involves creating a hash of the received message, using the signing partner's public key to decrypt the hash sent with the original message, and verifying that both hash values are equal.

### Certificate Validation

PingFederate always checks certificates to see if they have expired, both when they are initially imported and at runtime when they are used to encrypt, decrypt, and digitally sign or verify assertions.

PingFederate can also check to see whether a certificate has been revoked, using either Certificate Revocation Lists (CRLs) or the Online Certificate Status Protocol (OCSP). Depending on the content of the certificate in question and your requirements, the server will perform either of these checks during SSO or SLO processing for the following cases:

- Signature verification
- Validation of a client certificate used for authentication to PingFederate when the server is handling direct client requests
- Validation of the server SSL certificate when PingFederate is acting as the client making an HTTPS request to a separate server

If a certificate is expired or revoked, the associated SSO or SLO transaction fails at runtime and an error is written to the transaction log. In the administrative console, an expired or revoked certificate is identified as such in the Status column of its respective Certificate Management list.

#### CRL Revocation Checking

This process involves querying a CRL distribution-point URL and ensuring that a certificate is not on the returned revocation list maintained at the site. The URL is specified in the certificate.

No setup is needed in the administrative console to enable CRL checking. PingFederate automatically checks CRLs if all of the following conditions are met:

- The certificate contains the URL where the CA maintains its CRL.
- The URL is accessible.
- The returned CRL is signed and the signature verified.
- CRL validation is not explicitly disabled as a failover option in the OCSP setup (see “Certificate Revocation Checking” on page 97).

#### OCSP Revocation Checking

OCSP was developed as an alternative to CRL validation and provides a more centralized and potentially more reliable means of checking certificate status. In this scenario, an OCSP Responder URL is normally embedded in the incoming certificate (a configured default URL may be used, alternatively). The URL, maintained by the issuing CA, is used to query the certificate status.
The primary difference between OCSP and CRL checking is how the verification occurs. CRL checking requires the requesting client to determine if the certificate has been revoked (or if any of the certificates in the chain of issuer certificates has been revoked), based on the returned CRL. With OCSP, the client sends the certificate itself, and revocation checking is handled by the Responder server, which returns the certificate status.

A PingFederate administrator can enable and configure OCSP processing in the administrative console (see “Certificate Revocation Checking” on page 97). The protocol may be used exclusively or in conjunction with CRL checking as a backup. For more information about OCSP, see www.ietf.org/rfc/rfc2560.txt.

Digital Signing Policy Coordination

To coordinate digital signature policy, partners must first agree about whether they will sign SAML messages or tokens. In some cases, the protocol specifications require signatures—for example, all SAML STS tokens and all SSO assertions sent across the POST binding must be signed. (These requirements are enforced by the PingFederate administrative console and the runtime protocol engine.) Other uses of the digital signatures are optional between partners and enforced if specified for a partner connection.

If a digital-signing certificate is not issued by a trusted CA (that is, “self-signed”), then the signing partner must send the public-key certificate out-of-band (for example, via email) to the partner. The partner must import the certificate into PingFederate when configuring a connection to the signing partner for SSO/SLO or STS.

If the certificate is signed by a trusted CA and the signing partner chooses to embed the certificate in all signed messages, then the verifying partner can elect to use the embedded certificate for signature verification, after validating it against the Subject DN of the original certificate. The public-key certificate may or may not be sent out-of-band (just the Subject DN is required).

Tip: PingFederate can extract the Subject DN from the certificate, when available, during the signature-verification configuration.

The next section provides more information about the two alternative signature-verification trust models described above, from the standpoint of the verifying partner.

Trust Models

For validating digital signatures, PingFederate provides a selection of trust models in the administrative console for each partner connection, based on the certificate categories listed below. Note that for each trust model, PingFederate always verifies that the certificate is current and that the signature in the message can be verified using the certificate specified. Additional checks depend upon the trust model selected.

- Anchored Certificate

  In this case, certificates used for signature verification must be issued by a trusted CA, and the certificate chain must be verifiable recursively back to the root issuer. PingFederate validates the certificate (including recursive revocation checking, when enabled, back to the issuer) for all signed messages from the partner.

  In addition, when this option is selected, the incoming message must include the verification certificate for the signature. PingFederate uses that certificate to verify signatures from the partner if its Subject DN matches the partner’s public certificate, as specified in the administrative console, and the issuer CA
certificate is part of the trusted store. This feature provides a dynamic trust model that overcomes the problem of interrupting service to change out expired certificates.

- **Unanchored Certificate**
  
  When this option is chosen, incoming signatures are verified exclusively using a specific certificate imported for a connection into PingFederate (or a secondary, backup certificate when specified). The certificate may be self-signed or issued by a trusted CA. The certificate chain, if any, is not verified. However, revocation checking, when enabled, is performed up any existing chain as far as available.

**Secure Sockets Layer**

SSL certificates signed by a CA can be used to identify one or both ends of the federation. SSL/TLS provides an encrypted connection between the two parties in which the content of a message is not exposed, thus ensuring confidentiality and message integrity.

**SAML SSL/TLS Scenarios**

SSL/TLS should be used in association with the SOAP responder URL and Single Sign-on Service located at an IdP site. On the SP side, the Artifact Resolution Service should also use SSL/TLS. Optionally, SSL/TLS may also be used to secure communication between internal user data stores and PingFederate and between the PingFederate STS and Web Service client or provider applications.

**Authentication**

Three methods of authentication, described below, are available for use with PingFederate for browser-based SSO to authenticate connection partners making SOAP requests. For SOAP authentication by STS clients, a separate option using either or both of the first two methods, may be configured (the third method, digital signing, is automatically required). The selection of one or more method(s) must be agreed upon between partners and synchronized within IdP and SP federation implementations:

- **HTTP Basic Authentication**: partners identify themselves by passing username and password credentials.
- **SSL Client Certificate Authentication**: partners use SSL Client Certificates presented during SOAP request transactions. Each partner needs to import the
other's certificate out-of-band (see “SSL Client Keys & Certificates” on page 91).

**Important:** The SSL/TLS server-client handshake involves negotiating cipher suites to be used for encryption/decryption on each side of a secured Internet transaction. PingFederate supports only stronger cipher suites; to enhance security, weaker cipher suites are commented out of two configuration files located in `<pf_install>/server/default/data/config-store`:
- com.pingidentity.crypto.SunJCEManager.xml
- com.pingidentity.crypto.LunaJCEManager.xml

To ensure the most secure transactions, we recommend that administrators retain this cipher-suite configuration.

Due to import control restrictions, the standard Java Runtime Environment (JRE) distribution supports strong but not unlimited encryption. For this reason, the strongest cipher suites in the same configuration files are also commented out. To use the strongest encryption, when permissible, remove the comments from the AES 256 cipher suites and download and install the appropriate version of “Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files” from the Sun download Web site (http://java.sun.com/javase/downloads).

- Digital Signatures: partners sign the XML message transmitted via the SSL/TLS connection. Signatures are verified by the receiver based upon the certificate(s) configured for that connection. Each partner should import the other's certificate(s) out-of-band (see “Digital Signing and Decryption Keys & Certificates” on page 93).

**Verifying Trusted Certificates**

PingFederate validates the trust of all certificates. A certificate is trusted if the certificate of its issuer is in PingFederate's trusted certificate store. The root certificate of the CA, by which a certificate is issued, must be imported into PingFederate's trusted certificate store or contained in the Java runtime `cacerts` store.

**XML Encryption**

PingFederate supports the optional SAML 2.0 specification allowing for encryption of assertions (including STS SAML tokens), which further enhances confidentiality when required.

For SAML 2.0 SSO connections you can choose to encrypt entire assertions, the user's name identifier, and/or other user attributes. You can use signature verification and signing keys to encrypt and decrypt messages, respectively.

**About PingFederate Express**

PingFederate Express™ is a lightweight SAML 2.0 endpoint for federation partners who want a quick and easy way for an SP to process SAML assertions sent from an IdP partner who is using PingFederate. Available separately for specific platforms,
PingFederate Express provides a convenient, highly automated approach to configuring end-to-end basic Internet SSO between partners.

For an IdP, PingFederate provides a built-in template that an administrator uses to configure Express connections. The template automatically configures most settings (see “Using the Express Connection Template” on page 186). Once the configuration is complete, the administrator exports the connection metadata into a ZIP file.

On the SP side, a PingFederate Express administrator uses the ZIP as part of an automated installation procedure to configure the corresponding IdP connection in a matter of minutes.

**Using Auto-Connect**

PingFederate allows organizations to provide secure Internet SSO on the fly—that is, without the need for configuring partner-specific, browser-based SSO connection parameters. This feature—Auto-Connect™—extends SAML 2.0 SP-initiated SSO or SLO and metadata specifications to enable deployments to retrieve partner connection information securely on an as-needed basis. (For information about SAML 2.0, see the “Supported Standards” chapter in *Getting Started*.)

The feature is especially useful to an SP who wants to provide SSO capability to more than one partner. A Software-as-a-Service (SaaS) provider, for example, can provide SSO to innumerable clients without specifying redundant connection information for each one. Auto-Connect can also help an enterprise, acting as an IdP, to provide easily scalable SSO for multiple outsourced services.

For either an IdP or SP PingFederate server, you can implement Auto-Connect for any number of partners by configuring a common initial setup and a list of domain names. For an IdP, the domain-name list contains SP partners from whom your site will accept Auto-Connect authentication requests. For an SP, the list contains IdP-partner domains to which your site can send authentication requests and receive SSO assertions.

For information about configuring Auto-Connect for your federation partners, see “Configuring SP Auto-Connect” on page 189 or “Configuring IdP Auto-Connect” on page 265.

**Providing Metadata**

You enable Auto-Connect as part of Server Settings from the Main Menu (see “Choosing Roles and Protocols” on page 64). Once Auto-Connect is enabled and the initial setup is fully configured and activated, partners can retrieve your connection metadata via HTTP. At runtime, Auto-Connect deployments at partner sites use the endpoints provided in the metadata to interact with your server and complete SSO or SLO processing.

The metadata, which follows SAML 2.0 specifications, must be signed, and the validity of the data is time-limited (see “Auto-Connect Security Model” on page 17) and “Configuring Auto-Connect Metadata Lifetime” on page 71).

**Runtime Processing**

Auto-Connect runtime processing starts when a user tries to reach a protected SP resource. The process depends on SP Web-application functionality that determines
the user’s IdP domain (for example, from a submitted email address) and passes it to
the SP PingFederate server in the SSO request.

Figure 1 and the accompanying “Processing Steps” describe the complete SSO
processing flow:

![Auto-Connect Processing Flow](image)

**Figure 1:** Auto-Connect Processing Flow

**Processing Steps**

1. Internet user sends a logon request with an email address to an SP application.
   For example:
   
   john@mycompany.com

2. The application parses the email address and sends a request to PingFederate.
   For example:
   
   https://sp_host.com:9031/sp/
   startSSO.ping/?Domain=mycompany.com

3. The SP PingFederate server looks up the domain in a list of domain names
   allowed to use Auto-Connect.

4. If the domain is in the list, the SP retrieves connection metadata from the IdP’s
   public endpoint.

   By default, PingFederate looks for the metadata by prepending
   http://saml to the domain. For example:
   
   http://saml.mycompany.com

   This default location can be changed, if necessary, in the Allowed Domains lists
   configured in the PingFederate administrative console.

5. After validating the metadata (see “Auto-Connect Security Model” on page 17),
   the SP sends an authentication request to the IdP’s SSO service.

6. If the request `<Issuer>` is not among the IdP’s static-connection partners, the
   IdP PingFederate server looks for the issuer’s domain name in the list of domains
   allowed to use Auto-Connect.

7. The IdP retrieves the SP’s metadata via its public endpoint and verifies the
   metadata signature.

   The process is the same as that used by the SP in **Step 4**.

8. The IdP requests user authentication via the configured adapter instance.
9. Once the user is authenticated, the IdP returns a signed SAML assertion to the SP’s Assertion Consumer Service (ACS) endpoint.

10. (Not shown) The SP logs the user on to the requested resource via the configured SP adapter.

**Auto-Connect Security Model**

Auto-Connect processing requires digital signatures to ensure the authenticity of the published metadata as well as all subsequent SSO or SLO requests and responses. The certificate used to sign the metadata is included in the metadata, and all certificates must be signed by a trusted Certificate Authority; thus, partners need not exchange certificates out of band.

In addition to validating certificates, the PingFederate runtime server compares the partner certificate with the entity ID (the “Issuer”) found in the SAML message. Then the server matches the entity ID against the configured list of allowed Auto-Connect domains.

Figure 2 illustrates the security validation process:

![Auto-Connect Security Model Diagram](image-url)

Figure 2: Auto-Connect Security Model

Note that the diagram assumes that the same certificate is used for signing both the metadata and the runtime SAML messages. This is convenient, but not required.

**User Provisioning**

PingFederate provides two different kinds of user provisioning for browser-based SSO, one designed for an IdP and one for an SP:

- At an IdP site, you can provision and maintain user accounts (including “deprovisioning”) at selected hosted-software providers (see the next section, “SaaS Provisioning”).

- At an SP site, you can provision accounts for your own organization automatically, using information from SAML assertions received during SSO events (see “Express Provisioning” on page 18).
### SaaS Provisioning

For IdP sites, PingFederate offers automated provisioning and deprovisioning to facilitate SSO to either (or both) of two SaaS providers: Google Apps and Salesforce.

*Tip:* SaaS Provisioning, including quick-connection templates for partner SSO, is available separately from Ping Identity. Contact sales@pingidentity.com for more information.

When SaaS Provisioning is enabled, the PingFederate runtime engine polls the IdP organization’s user store periodically. The server uses a separate database internally to monitor the state of the user store and keep user data synchronized between the organization and the hosted SaaS application (see Figure 3).

![Figure 3: SaaS Provisioning Processing](image)

PingFederate provides built-in support for Microsoft’s Active Directory and the Sun Directory Server (formerly Sun ONE) as user-data sources; templates are used to preconfigure many provisioning settings. Although these are the only data stores formally tested and supported, other LDAP data stores will likely work as well (see “Identifying the Source Data Store” on page 174). For convenience, PingFederate provides a sample template that can be used for other types of LDAP servers to simplify the provisioning configuration (see “Configuring an LDAP Connection” on page 77).

Tested internal data stores include Hypersonic, MySQL, and Oracle databases (a demonstration-only, embedded Hypersonic database is installed by default). Again, any relational database may be used—scripts are provided to aid setup (see “Configuring SaaS Provisioning Settings” on page 69).

*Tip:* Administrators have access to a command-line utility that can be used to monitor and make adjustments to the internal database as needed (see “SaaS Provisioning CLI” on page 49).

### Express Provisioning

At an SP site, PingFederate can create and update local user accounts in an external LDAP directory or Microsoft SQL Server 2005 as part of SSO processing—Express Provisioning. This feature allows SPs to maintain accounts for users that authenticate from IdP partners without having to provision accounts manually prior to a user’s first SSO.

When configured, the PingFederate SP server writes user information to the local user store using attributes from the incoming SAML assertion. For SAML 2.0 partner connections, assertion attributes can be supplemented with user attributes returned from an Attribute Query (see “Attribute Query and XASP” in the “Supported Standards” chapter of Getting Started).
PingFederate can also update existing user accounts. When this option is enabled, PingFederate can add or overwrite attributes for a local user account each time SSO for a user is processed. (Note that once user attributes are provisioned, they cannot be removed using Express Provisioning.)

For information about enabling Express Provisioning, see “Choosing Connection Options” on page 210. For configuration information, see “User Provisioning” on page 244.

**Federation Planning Checklist**

An essential first step in establishing an identity federation involves discussions and agreements between you and your connection partners. The sections below comprise a partial checklist of items that should be coordinated before you deploy PingFederate.

**Tip:** For basic SSO deployment between partners, PingFederate Express virtually eliminates the need for much of the coordination described in this section, in addition to simplifying and expediting the configuration for both an IdP and an SP (see “About PingFederate Express” on page 14).

**Tip:** Extensive coordination and configuration may also be avoided by using Auto-Connect (see “Using Auto-Connect” on page 15).

**Standards and Specifications**

Choose which federation protocol(s) your deployment will support. For SAML SSO configurations, decide which profiles and bindings will be used. (See the “Supported Standards” chapter in Getting Started.)

**Signing and Validation**

Decide which SAML messages—assertions, responses, requests—will be digitally signed and how the messages will be verified by your federation partner. If messages are signed, decide how certificates will be exchanged (for example, secure email). (See “Certificates, SSL, and XML Encryption” on page 10.)

**Back-Channel Security**

Determine what type of SOAP channel authentication will be used: Basic or SSL/TLS. If SSL/TLS is used, determine whether server-only or both server and client certificates will be needed and how they will be managed. Also decide what level of security will be required for connections to back-end data stores or identity management systems.

**Trusted Certificate Management**

Determine whether both partners are using SSL/TLS and/or signing certificates that have been signed by a major CA. (If self-signed certificates or nonstandard CAs are used, the signed certificates must be exchanged and imported into Trusted Certificate stores.) Also, determine whether you want to adopt a trust model that uses embedded certificates (see “Trust Models” on page 12).
Deployment

Decide how PingFederate fits into your existing network. Also, determine whether high-availability and/or failover options are required (see the PingFederate Server Clustering Guide).

Federation Server Identification

Determine how you and your partner(s) will identify your respective federation deployments. Under federation standards, both the sender (IdP) and the receiver (SP) of an assertion must be uniquely identified within the identity federation (see “Configuration Data Exchange” on page 21).

With PingFederate, you define a unique ID for each supported protocol (see “Specifying Federation Information” on page 67). Optionally, you can also use Virtual Server IDs on a connection-by-connection basis. This option provides more configuration flexibility in cases where you need more than one connection to the same partner for different purposes. For example, you would want to use virtual IDs if you are an IdP and you have an SP partner who requires a different set of attributes to launch different applications. Assigning virtual IDs allows you to configure multiple connections to such a partner, each set up to manage attributes differently. (Note that the partner must also have a federation deployment that supports multiple federation IDs.)

You can assign virtual server IDs either as an IdP during configuration of an SP connection (see “General Information” on page 117) or as an SP configuring an IdP connection (see “General Information” on page 212).

Tip: PingFederate also provides for virtual host names, which differ from virtual IDs (but are not mutually exclusive); they are intended to be used when your network configuration is such that you receive federation messages under more than one domain name (see “Using Virtual Host Names” on page 42).

Server Clock Synchronization

Ensure that both the SP and IdP server clocks are synchronized. SAML messages and STS tokens provide a time window that allows for small synchronization differentials. However, wide disparities will result in assertion or request time-outs.

User Data Stores

Identify the type of data store that contains user data when needed: LDAP, JDBC, or Custom (see “Data Stores” on page 9).

Web Application and Session Integration

Decide how PingFederate as an IdP receives subject identity information, either from an STS token or a user session.

For an SP, decide how PingFederate will forward user identity information to the destination Web application or system to start a session.

(See “SSO Integration Kits and Adapters” on page 4 and “Token Processors and Generators” on page 3.)

Transaction Logging

PingFederate provides basic transaction logging and monitoring. Decide whether transaction logging should be integrated with a systems management application and
whether you have regulatory compliance requirements that affect your logging processes. (For more information, see “Log File Generation” on page 24.)

Identity Mapping

For browser-based SSO, decide whether you will use PingFederate to link accounts on your respective systems using a persistent name identifier, or whether you will use account mapping (see “Identity Mapping” on page 5).

Attribute Contract Agreement

If your federation partnership will not use account linking, or will not use it exclusively, then you and your partner must agree on a set of attributes that the IdP will send in an assertion for either SSO or Web Service access. (For more information, see “Attribute Contracts” on page 7.)

Metadata Exchange

If you are using SAML, decide whether you will use the metadata standard to exchange XML files containing configuration information. PingFederate makes it easy to use this protocol, which provides a significant shortcut to setting up your partner connections. (If your partner is also using PingFederate or supports standards permitting runtime metadata exchange, the process can be even simpler—see “Using Auto-Connect” on page 15. The process is also simplified and mostly automated using PingFederate Express—see “About PingFederate Express” on page 14)

Configuration Data Exchange

If your partner's deployment does not produce or consume a metadata file that conforms to SAML metadata specifications, you may need to exchange connection information manually. The following sections list some common configuration details that must be exchanged if metadata files are not used. (These lists are not exhaustive.)

IdP to SP

If you are the IdP, your SP partner will need some or all of the following connection information (depending upon which profiles and bindings you are configuring):

- **Unique ID**—Identifies the IdP that issues an assertion or other SAML message. For SAML 2.0, the ID is the IdP's Entity ID; for SAML 1.x, it is the IdP's Issuer; for WS-Federation, it is the IdP's Realm.
  
  PingFederate also supports the optional use of virtual IDs (see “Federation Server Identification” on page 20).

- **SOAP Artifact Resolution URL**—The endpoint your site uses to receive an SP's SOAP requests when the artifact binding is used.

- **Single Logout Service URL**—The destination of SLO request messages.

- **Single Sign-On Service URL**—The endpoint where you receive and process assertions.

SP to IdP

If you are the SP, your IdP partner will need some or all of the following connection information (depending upon which profiles and bindings you are configuring).
Chapter 1

Key Concepts

- **Unique ID**—Identifies the SP. For SAML 2.0, the ID is the *Entity ID*; for SAML 1.x, it is the SP’s *Audience*; for WS-Federation, it is the SP’s *Realm*.

  PingFederate also supports the optional use of virtual IDs (see “Federation Server Identification” on page 20).

- **SOAP Artifact Resolution Service URL**—The endpoint to use for SOAP requests when the artifact binding is used.

- **Single Logout Service URL (SAML 2.0)**—The destination of SLO request messages.

- **Assertion Consumer Service URL**—The location where the SP receives assertions.

- **Target URLs**—The URLs for the protected resources that a user is trying to access.

**Mutual Settings Between Parties**

Many settings must be mutually set by the parties. This information might include such items as:

- **Attributes**—User information that will be sent in an assertion, if any (see “About Attributes” on page 6).

- **Signing certificates**—The SAML and WS-Federation protocols specify a number of conditions under which digital signatures are either required or optional (these conditions are built into the PingFederate connection-setup screens).

- **SOAP connection type and authentication style**—For SAML connections using the back channel (using the artifact binding, for example), HTTP Basic authentication, SSL client certificate authentication, digital signatures, or some combination of the three is required. You and your partner must exchange the necessary credentials, certificates, and/or signing keys.
System Administration

This chapter describes general administrative functions for PingFederate, including:

- “Starting and Stopping PingFederate” on page 24
- “Log File Generation” on page 24
- “Exporting Metadata” on page 27
- “Signing XML Files” on page 31
- “Using the Configuration Archive Utility” on page 32
- “Account Management” on page 34
- “Alternative Console Authentication” on page 38
- “Managing Email Configuration” on page 40
- “Using Virtual Host Names” on page 42
- “Changing Configuration Parameters” on page 42
- “Installing a New License Key” on page 45
- “Automating Configuration Migration” on page 46
- “SaaS Provisioning CLI” on page 49
- “Customizing User-Facing Screens” on page 52

Note: The information in this chapter is presented from the viewpoint of an administrative user with “Admin” permissions (see “Account Management” on page 34).
Starting and Stopping PingFederate

(Windows)

To start PingFederate:

- From Start > Run dialog or a command prompt, run the batch file:
  
  `<pf_install>\pingfederate\bin\run.bat`

  Or:
  
  Open the `\bin` folder in Windows and double-click the file.

  Wait a moment for the script to execute. The server is started when you see the message “Started in [xx]:[yy]ms” in the command window, near the end of the start-up sequence.

To shut down PingFederate:

1. Enter Ctrl+C in the command-prompt window.
2. Enter y to terminate when prompted.

(Unix/Linux)

To start PingFederate:

1. From a command prompt, change directories to `<pf_install>/pingfederate/bin`.
2. Execute the `run.sh` file.

  Wait a moment for the script to execute. The server is started when you see the message “Started in [xx]:[yy]ms” in the command window, near the end of the start-up sequence.

To shut down PingFederate:

- Enter Ctrl+C in the terminal window.

(All Platforms)

To access the PingFederate administrative console:

- Launch a Web browser and go this location:
  
  `https://<DNS_NAME>:<port>/pingfederate/app`

  where `<DNS_NAME>` is the fully qualified name of the machine running the PingFederate server and `<port>` is the port where the administrative console listens. The default port is 9999.

Log File Generation

PingFederate generates log files that document the system’s activities. The logs are stored in the `<pf_install>pingfederate/log` directory and include:

- `admin.log` — Records all actions performed by administrative console users (see “Administrator Audit Logging” on page 25)
- `transaction.log` — Records individual identity-federation runtime transactions at specified levels of detail

The level of detail is configurable globally or on a connection-by-connection basis (see “Runtime Transaction Logging” on page 26).
Log File Generation

- `server.log` — Records all PingFederate runtime and administrative server activity
- `provisioner.log` — Records only SaaS Provisioning activity (see "SaaS Provisioning" on page 18)

**Tip:** PingFederate logs user attributes, when they are present, in the server log, the transaction log, or both. When privacy is required for sensitive user attributes, you can configure PingFederate to obfuscate (mask) their values in the server and transaction logs (see "Attribute Masking" on page 9).

Other logs contained in the `pingfederate/log` directory are generated by the PingFederate Web container. These logs, `<date>.request.log`, record all HTTP requests for the given date.

**Note:** Properties controlling request logging are contained in the Web-container configuration file `jboss-service.xml` located in the PingFederate-installation directory:

`pingfederate/server/default/deploy/jetty.sar/META-INF`

In addition, a JBoss-generated start-up log, `boot.log`, is located in the directory:

`pingfederate/server/default/log`

The PingFederate-generated logs can be controlled through the `log4j.xml` file located in `pingfederate/server/default/conf/`. See comments in the file for more information. Refer to the `log4j` open-source project for more information about logging levels and other configuration parameters (http://logging.apache.org/log4j/docs).

By default, PingFederate installs with a highly verbose level of logging. However, verbose logging may have a performance impact and clutter the log files. You may choose to lower the level, but we recommend that you not set it below `WARN`. For the `transaction.log`, note that any setting below `INFO` turns logging off.

**Important:** The `transaction.log`, the `admin.log`, and the `provisioner.log` files roll over at midnight each day. The system keeps all of the resulting historical log files. The `transaction.log` can become quite large, depending on your production load and settings (see "Runtime Transaction Logging" on page 26); you might wish to back up or remove older files on a routine basis.

Other PingFederate log files roll over when they reach 10MB. The system keeps five old log files of each type before overwriting the oldest. (This number can be changed in the `/conf/log4j.xml` file.)

The following sections provide more detail about the `admin` and `transaction` logs.

**Administrator Audit Logging**

PingFederate records actions performed by server administrators. This information is recorded in the `admin.log` file. While the events themselves are not configurable, `log4j.xml` configuration settings may be adjusted to deliver the desired level of detail surrounding each event.
Each entry in the `admin.log` file is on a separate line and represents a single administrator action. The general format of each entry is the same, though specific events are recorded with information relevant to each type. Events are recorded when the corresponding `Save` button in the administrative console is clicked.

A log entry is generated for each of the events listed below:

- Password change
- Password reset
- Account activation
- Account deactivation
- Role change
- Login attempt
- Explicit user logouts (no time-outs)
- Data store created
- Data store modified
- Data store deleted
- Certificate management
- SP connection created, modified, or deleted
- IdP connection created, modified, or deleted
- URL-to-adapter mapping management
- SP Adapter created, modified, or deleted
- IdP Adapter created, modified, or deleted
- Server settings management
- Metadata export
- Configuration archive
- IdP Discovery management
- SP Affiliation created, modified, or deleted
- Attribute requester mapping
- IdP default URL modified
- SP default URLs modified
- XML file signatures applied

Each log entry contains information relating to the event, including:

- The time the event occurred on the PingFederate server.
- The username of the administrator performing the action.
- The role(s) assigned to the administrator at the time the event occurred.
- The type of event that occurred.
- Details about the event.

Each of the above fields is separated by a vertical pipe (|) for easier parsing.

## Runtime Transaction Logging

PingFederate provides for flexible, scalable logging of all federated-identity transactions (inbound and outbound XML messaging). Transaction logging can be configured to any of four modes on a connection-by-connection basis (see “General Information” in either the IdP or SP “SSO Configuration” chapters).
You also have the option of overriding transaction logging for all connections (to find this feature, click the relevant Manage All . . . link under IdP/SP Connections on the Main Menu). You might wish to use this override for troubleshooting or as a one-step means of raising or lowering all connection logging modes to the same level.

**Transaction Logging Modes**

The table below describes the four transaction logging modes:

<table>
<thead>
<tr>
<th>Mode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>No transaction logging.</td>
</tr>
<tr>
<td>Standard</td>
<td>(Default) Logs summary information for each transaction message, including:</td>
</tr>
<tr>
<td></td>
<td>• Timestamp</td>
</tr>
<tr>
<td></td>
<td>• Hostname:Port</td>
</tr>
<tr>
<td></td>
<td>• LogMode</td>
</tr>
<tr>
<td></td>
<td>• ConnectionID</td>
</tr>
<tr>
<td></td>
<td>• SAML Status Code&gt; (for SAML responses only)</td>
</tr>
<tr>
<td></td>
<td>• Context</td>
</tr>
<tr>
<td></td>
<td>• MessageType</td>
</tr>
<tr>
<td></td>
<td>• SAML ID (for SAML messages only)</td>
</tr>
<tr>
<td></td>
<td>• Endpoint (for outbound messages only)</td>
</tr>
<tr>
<td></td>
<td>• Target URL (if SSO transaction)</td>
</tr>
<tr>
<td>Enhanced</td>
<td>Includes everything logged at the Standard level plus:</td>
</tr>
<tr>
<td></td>
<td>• SAML_SUBJECT*</td>
</tr>
<tr>
<td></td>
<td>• Binding</td>
</tr>
<tr>
<td></td>
<td>• RelayState (if available)</td>
</tr>
<tr>
<td></td>
<td>• SignaturePolicy</td>
</tr>
<tr>
<td></td>
<td>• SignatureStatus</td>
</tr>
<tr>
<td></td>
<td>• HTTP Request Parameters (outbound messages only)</td>
</tr>
<tr>
<td></td>
<td>* Only when available in a SAML assertion, a single-logout request, a Request Security Token Response (RSTR), or an authentication request (AuthnRequest)</td>
</tr>
<tr>
<td>Full</td>
<td>Includes everything logged at the Enhanced level plus the complete XML message for every transaction.</td>
</tr>
</tbody>
</table>

Each of the above fields is separated by a vertical pipe (|) for easier parsing.

**Exporting Metadata**

For SAML deployments PingFederate supports the export and import of metadata files, which federation partners can use to expedite their configuration. You can export metadata for any connection (or select certain nonconnection-specific metadata for export) via the Main Menu. You can import your partner’s metadata file, when available, at the beginning of the connection-configuration process (see...
“Managing IdP Connections” on page 206 or “Managing SP Connections” on page 110).

Tip: Connection-metadata export is also available from the Manage Connections screens, which can be reached from the Main Menu via the Manage All . . . link under either SP Connections or IdP Connections.

Tip: IdP administrators can use the Manage Connections screen to export metadata for PingFederate Express SP connections in a one-step process (see “Using the Manage Connections Screen” on page 111). (For information about Express connections, see “About PingFederate Express” on page 14.)

To reach the Metadata Export task:

► Click Metadata Export under Administrative Functions on the Main Menu.

To export connection metadata:

1. If your PingFederate server is configured to act as both an IdP and an SP, indicate which type of configuration you will export and click Next.

2. On the Metadata Mode screen, choose the option to “Use a connection . . .” and click Next.

3. On the Connection Metadata screen, select the connection from the drop-down menu and click Next.

4. (Optional) On the Metadata Signing screen, select a certificate to use for signing the metadata XML file and click Next.

Note: If you want to include the public-key information in the signed XML file, select the Key Info option.

For more information, see “Signing XML Files” on page 31.

For SP connections using PingFederate Express, this screen does not appear; Express metadata is not signed.

5. On the Export & Summary screen click the Export button, save the file, and then click Done.

To export selected metadata:

1. If your PingFederate server is configured to act as both an IdP and an SP, indicate which type of configuration you will export and click Next.

2. On the Metadata Mode screen, select the option to “Select information . . .” and click Next.
3. If you support more than one federation protocol, select the desired protocol on the Protocol screen and click Next.

4. Configure any or all of the remaining steps in the task (click Next to skip steps). For information see:
   - “Defining Metadata Attribute Contracts” on page 29.
   - “Choosing a Metadata Signing Key” on page 30.
   - “XML Encryption Certificates” on page 30.

5. (Optional) On the Metadata Signing screen, select a certificate to use for signing the metadata XML file and click Next.

   Note: If you want to include the public-key information in the signed XML file, select the Key Info option.

   For more information, see “Signing XML Files” on page 31.

6. On the Export & Summary screen click the Export button, save the file, and then click Done.

Defining Metadata Attribute Contracts

The Attribute Contract screen allows you to define the attribute contract you want to export in the metadata. For more information, see “Attribute Contracts” on page 7.

To reach this screen:
1. Click SAML Metadata Export under Administrative Functions on the Main Menu.

2. On the Metadata Mode screen, click the button for selecting the information manually and click Next.

To add an attribute:
1. Enter an attribute on the Attribute Contract screen and click Add.

2. Continue to add attributes as needed and click Next.

To edit an attribute name:
1. Click Edit and make your change.

2. Click Update.

To delete an attribute:

   ▶ Click Delete.
Choosing a Metadata Signing Key

In a metadata file you can manually include the public key for partners to use to verify the digital signature you will use to sign SAML messages. For more information, see “Digital Signing and Decryption Keys & Certificates” on page 93.

To export your public signature verification key:

▲ Select the key from the drop-down list and click Next.

XML Encryption Certificates

In your metadata file you can manually include the XML encryption key and certificate your partners can use to encrypt SAML messages.

To export an XML encryption key:

▲ Select the key from the drop-down list and click Next.

If the certificate is not shown, click Manage Certificates to import it.

Completing the Export

On the Export & Summary screen, you can complete the XML-file download or change any information by clicking any of the headings in the Summary.

Important: To finish the download, you must click the Export button at the bottom left of the Export Metadata screen.
PingFederate supports digital signing of SAML metadata files or any other XML files that you and your partner might want to exchange. A signature applied to an XML file ensures that the file is from the original source and that its contents have not been modified by a third party.

When you configure a partner connection, you can also verify and import signed metadata files. For information:
- As an SP configuring an IdP connection, see “Importing Metadata” on page 211.
- As an IdP configuring an SP connection, see “Importing Metadata” on page 116.

XML file signing is available from the Main Menu under Administrative Functions.

**To sign an XML file:**
1. On the Select XML File screen, locate and open the file.
2. On the Digital Signature Settings screen, choose the certificate containing your signing key from the drop-down list.

   ![Digital Signature Settings Screen](image)

   - **Note:** By default, certificate and public-key information is included in the signed XML file. If you do not wish to include this information, clear the Key Info checkbox.

3. On the Export & Summary screen, click the Export button to save the signed file.

   ![Export & Summary Screen](image)

   - **Important:** Be sure to click Export in the lower-left portion of the Export & Summary screen; clicking Done does not complete the operation.
Using the Configuration Archive Utility

PingFederate's archive utility allows you to export the current administrative-console configuration to a ZIP file and to import an existing archive for immediate deployment into a running PingFederate server.

**Tip:** A time-stamped configuration archive is created as a backup automatically every time you log on to the administrative console and before an existing archive is imported. The archives are stored in pingfederate/Server/default/data/archive.

**Tip:** PingFederate includes a separate command-line utility that provides a finer level of control over configuration migration, as well as providing for scripting of routine configuration-management tasks (see “Automating Configuration Migration” on page 46).

---

Configuration archives can be used as backups or to transfer data from one PingFederate server to another, including migration to a new PingFederate release.

**Important:** Draft connections in archives are not imported. Complete any unfinished partner connections if you wish to include them in a full backup archive or in an archive to be used for configuration migration.

**Note:** The archive utility is intended for administrative-console configuration data only. It does not include error-page or other end-user HTML templates (see “Customizing User-Facing Screens” on page 52), nor any files under <pf_install>/server/default/conf. If any changes have been made to the default templates or configuration files, you must copy them over to new installations or other instances of PingFederate (assuming the changes are applicable).

The archive also does not capture license files, adapter JAR files, database drivers, or any other plug-ins; these also must be copied into any new instance of PingFederate. The import utility checks for and reports on any missing components (see “Using the Archive Import Screen,” next).

**To reach this screen:**

- Click **Configuration Archive** on the Main Menu.
To create an archive:
1. On the Select Import/Export screen, ensure Export is selected and click Next.
2. On the Export screen, click Export, save the download to your file system, and click Done.

To import and deploy an archive:
1. On the Select Import/Export screen, click Import and then Next.
   See the next section, “Using the Archive Import Screen,” for more information.

Note: Alternatively, you can deploy an archive manually, using the procedure below.

Caution: Deploying a configuration archive, either manually or by using the administrative console, always overwrites all existing configuration data.

To deploy an archive manually:
1. Copy a previously stored archive into the directory:
   `<pf_install>/pingfederate/server/default/data/drop-in-deployer`
2. Rename the copied file to `data.zip`.
   When the PingFederate server is running, the file is again renamed with a timestamp after a moment and the data automatically deploys, replacing the current UI configuration. Restarting the server is not required.
3. Consult the PingFederate server start-up window, or the server log file, for any messages concerning missing plug-in components or other errors.

Note: A data archive imported via the `drop-in-deployer` directory is deployed by default regardless of errors, unlike the default behavior using the Import screen.

Using the Archive Import Screen

When you initiate deployment of a configuration archive using the Import screen, PingFederate displays error messages if there are any missing plug-in components (such as adapters, database drivers, or token translators) on which the archive depends, or any mismatches of PingFederate licensing authorization.
If there are missing components or license inconsistencies, the import is halted by default to allow you to install the necessary components or license. However, you can choose to force the deployment and then install the necessary files later.

**Note:** Installation of any missing database drivers or other third-party libraries will require a PingFederate server restart.

**To deploy a configuration archive:**

1. On the Import screen, click **Browse** to locate the required ZIP file.
2. (Optional) Select **Force Import** to deploy the archive regardless of whether dependencies are detected.
3. Click **Import**.
4. Read the on-screen caution statement and indicate whether you want to continue.
   - When you click **Yes**, a message is displayed indicating the deployment result.
5. Click **Done**.

### Account Management

PingFederate provides a choice of single- or multi-user system administration (see “Setting Administration Options” on page 58).

**Note:** This choice is not presented if you are using your network’s LDAP user store or client certificates for authentication to the PingFederate administrative console (see “Alternative Console Authentication” on page 38).

If you choose native multi-user administration or if you are using alternative authentication, PingFederate provides role-based access control, as shown in Table 3. For native multi-user administration, you can choose to use email for password setting and resetting notifications.
When Auditor is assigned, no other roles may be set. Admin users may have multiple roles set.

**Tip:** The same user may log on from more than one browser or location. Also, by default, more than one user can log on to PingFederate at a time. You can change this default to restrict the administrative console to one administrative user at a time (see “Changing Configuration Parameters” on page 42).

Any number of auditors may log on at any time, regardless of the property setting.

**Note:** For security, after three failed logon attempts from the same location within a short time period, the administrative console will temporarily lock out further attempts by the same user. The user must wait one minute to try again.

**To reach the Account Management screen:**

- Click Account Management under Administrative Functions on the Main Menu.

**Note:** If you are using alternative PingFederate authentication, the Account Management configuration is not available. Set PingFederate-specific permissions for users in configuration files located in `<pf_install>/pingfederate/bin` (see “Alternative Console Authentication” on page 38).

---

**Table 3: PingFederate User Access Control**

<table>
<thead>
<tr>
<th>Role Assignment</th>
<th>Access Privileges</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Admin</td>
<td>Manage users, select administration style (single- or multi-user), define email notification policies, and configure an SMTP server connection. (This role is not provided if you are using LDAP authentication for administrative logon, since user management is handled outside of PingFederate.)</td>
</tr>
<tr>
<td>Admin</td>
<td>Configure partner connections and most system settings (except user management and local key/certificate handling).</td>
</tr>
<tr>
<td>Crypto Admin</td>
<td>Manage local keys and certificates.</td>
</tr>
<tr>
<td>Auditor</td>
<td>View-only permissions for all administrative functions.</td>
</tr>
</tbody>
</table>
Users with User Admin permissions can add other users, assign them any role, or reset their passwords, as well as change their own passwords. Other types of users can change only their own passwords from this screen (see “Changing Passwords” on page 38).

To add a user:
1. Click Create User.
2. On the User Information screen, enter the required fields (indicated by asterisks).

   **Note:** Only Username is required, unless you elected on the Account Management screen to have PingFederate send passwords via email, in which case you must supply an email address.

3. (Optional) Enter additional information.
4. Click Next to set up a password (see “Setting or Resetting Passwords” on page 37).

   **Important:** After you set the password and return to the Account Management screen, you must select permissions for the new user and click Save to complete the process.

To define a user’s permissions:
- Select or clear the checkboxes under the permission categories you want to assign or remove (see Table 3 on page 35).

   Clicking the Auditor button deactivates other permission selections.

   **Note:** For traceability and accountability purposes, users cannot be deleted; their records are retained and they can be reactivated if needed.

To enable password notification:
1. Click the password-notification checkbox.
2. If you have not yet configured PingFederate to use your email server, click Email Server Settings and complete the configuration (see “Managing Email Configuration” on page 40).

3. Click Save (or Next if you are installing PingFederate).

**Note:** If you are setting up email notifications for the first time, you must click Email Server Settings and configure the settings. If you are not sure of the correct settings, enter placeholders on the Email Notification screen; you can return later and update the information.

### Setting or Resetting Passwords

A user administrator can generate or assign temporary passwords for other users, either during user setup or at a later time (for example, if a user forgets his or her password).

**Note:** If you are using an LDAP user-data store for PingFederate authentication, password management is handled at the network level.

Initial or reset passwords may be used only once; the administrative console requires the user to change the password immediately after logging on.

<table>
<thead>
<tr>
<th>Configuring My Server</th>
<th>Help</th>
<th>Support</th>
<th>About</th>
<th>Logout</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Account Management</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>User Information</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>†User Information</td>
<td>✔</td>
<td>Password Generation</td>
<td>✔ Summary</td>
<td></td>
</tr>
</tbody>
</table>

This system generated password is valid for only a single use. Upon subsequent login, the user will be required to change his/her password.

[Generate one-time password]

[Enter password]

**To reach this screen:**

1. Click Account Management on the Main Menu.
2. Click Reset Password under Action for a user.

**To set or reset a user’s password:**

1. Either:
   - Click Generate one-time password.
   - Or:
   - Enter a password in the text box (no restrictions apply for a temporary password).
2. Click Done.

**Important:** The password and any other changes, including new user records, are not stored until you click Save on the Account Management screen.
After you click **Save** on the Account Management screen, the new password is emailed to the user automatically, if you have enabled email notifications (see “Account Management” on page 34).

## Changing Passwords

Any user can change his or her own password. For information about resetting another user’s password (if you are a user administrator), see the previous section.

**Note:** If you logged on to PingFederate using your network ID and password, you can change your password only at the network level. The new password will apply to PingFederate automatically the next time you log on.

To change your password:

1. Click **Account Management** on the Main Menu.
2. On the Account Management screen, click **Change Password** under the Action column.
3. Enter your **Current Password** and **New Password** (plus confirmation) and click **Save**.

Passwords must be at least six characters long and contain at least one upper-case character, one lower-case character, and one number.

**Important:** If you are the sole user administrator, take steps to ensure that you do not forget your new password.

## Alternative Console Authentication

You can configure PingFederate to use either your network's LDAP user-data store or client certificates for authentication to the administrative console, as an alternative to using PingFederate's own internal data store. You can configure either of these alternatives at any time.

Note that most user-management functions are handled outside the scope of the PingFederate administrative console when either LDAP or certificate-based
Alternative Console Authentication

authentication is enabled. Authorization levels, however, as described in “Account Management” on page 34, must be assigned in PingFederate configuration files.

Using LDAP Authentication

The LDAP authentication setup is available via configuration files in the <pf_install>/pingfederate/bin directory.

To configure PingFederate to use network LDAP authentication:

1. In the pingfederate/bin/run.properties file, change the value of pf.console.authentication as shown below:
   
   pf.console.authentication=LDAP

   **Note:** You can restore internal user-management control at any time by returning the value to native and restarting the PingFederate server.

2. In the pingfederate/bin/ldap.properties file, change property values as needed for your LDAP network configuration.
   
   See the comments in the file for instructions and additional information.

   **Important:** Be sure to assign LDAP users or designated LDAP groups (or both) to at least one of the PingFederate administrative roles as indicated in the properties file. For information about permissions attached to the PingFederate roles, see “Account Management” on page 34.

Using Certificate-Based Authentication

To enable client-certificate authentication, PingFederate administrative users must have imported into their Web browsers an X.509 key and certificate suitable for user authentication. In addition, the corresponding root CA certificate(s) must be contained in the Java runtime or the PingFederate trusted store (see “Trusted Certificate Authorities” on page 87).

Other setup steps, including designating user permissions, are required via configuration files in the <pf_install>/pingfederate/bin directory.

To enable certificate-based PingFederate console authentication:

1. If not already done, import the necessary client key and certificate into the Web browser used to access PingFederate.
   
   Refer to the browser's documentation, as needed, for instructions.

2. Log on normally (using username and password) to the PingFederate console as a user with permissions that include the role Crypto Admin.
   
   The default administrator includes this role. (For more information, see “Account Management” on page 34.)
3. Ensure the client-certificate's root CA and any intermediate CA certificates are contained in the trusted store (either for the Java runtime or PingFederate, or both).

To import a certificate, click **Trusted CAs** in the Security section under My Server.

**Tip:** You may wish to click the Serial number and copy the Issuer DN to use at Step 5.

4. In the `pingfederate/bin/run.properties` file, change the value of `pf.console.authentication` as shown below:
   ```
   pf.console.authentication=cert
   ```

5. In the `pingfederate/bin/cert_auth.properties` file, enter the Issuer DN for the client certificate as a value for the property:
   ```
   rootca.issuer.x
   ```
   where `x` is a sequential number starting at 1 (see the properties file for more information).

   If you copied the Issuer DN at Step 3, paste this value.

6. Repeat the previous step for any additional CAs as needed.

7. Enter the certificate user's Subject DN for the applicable PingFederate permission role(s), as described in the properties file.

   See “Account Management” on page 34 for more information about PingFederate permissions.

8. Repeat the previous step for all users as needed.

**Note:** Other settings in the properties file are used to display the user’s ID (the Subject DN) in abbreviated form in the administrative console.

9. Restart the PingFederate server (see “Starting and Stopping PingFederate” on page 24).

### Managing Email Configuration

If you are using email notification for password resets, licensing events, or certificate-expiration warnings, you must set up and maintain a connection to the email server that PingFederate will use to send messages (see “Account Management” on page 34 and “Configuring Runtime Notifications” on page 59).
To configure access to your email server:
1. Enter information into all fields. (Username and Password are not required.)

### Field Descriptions

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>“From” Address</td>
<td>The email address that appears in the “From” header line in email messages generated by PingFederate. The address must be in valid format but need not be set up on your system.</td>
</tr>
<tr>
<td>Email Server</td>
<td>The IP address or hostname of your email server.</td>
</tr>
<tr>
<td>SMTP Port</td>
<td>The SMTP port on your email server (default: 25).</td>
</tr>
<tr>
<td>Username</td>
<td>Authorized email username.</td>
</tr>
<tr>
<td>Password</td>
<td>User password.</td>
</tr>
<tr>
<td>Confirm Password</td>
<td>Re-entered password.</td>
</tr>
<tr>
<td>Test Address</td>
<td>(Optional) An email address where you want to send a test message.</td>
</tr>
</tbody>
</table>

To reach this screen:

- Click Email Configuration on the Main Menu under Administrative Functions.

  If this link is not displayed, then no email notifications are configured (see “Configuring Runtime Notifications” on page 59 or “Account Management” on page 34).

To configure access to your email server:
2. (Optional) Enter an email address (or addresses) in the Test Address field and click **Test Email Connectivity**.

A message next to the button indicates a successful test. Verify that the test email address received a message from the server.

Test reports are also written to the server.log file in the /log directory.

Using Virtual Host Names

In certain contexts, the SAML specifications require that XML messages include a URL identifying the host name to which the sender directed the message. (The name of the XML element containing the URL varies among protocols.) In addition, the recipient must verify that the value matches the location where the message is received.

Depending on your networking requirements, this specification can present problems—for example, in the case of proxy forwarding, where the final destination host name might be unknown to your federation partner. To provide more flexibility in such cases, you can set up a list of alternative host names for PingFederate to use as part of its message-security validation.

Note that virtual host names are used for a different purpose than virtual server IDs, which provide separate unique identifiers for a federation deployment, normally in the same domain (see “Federation Server Identification” on page 20). Depending on your needs, however, you can configure virtual server IDs and virtual hosts in the same installation of PingFederate.

Changing Configuration Parameters

PingFederate’s default administrative-console and runtime behavior is controlled in part by configuration properties contained in the file run.properties, located in: `<pf_install>/pingfederate/bin`. Table 4 describes the properties; refer to the file itself for default settings not specified here.

You can change these settings as needed. Restart the PingFederate server for changes to take effect.

**Note:** Properties related to server clustering are described in the PingFederate **Server Clustering Guide**.
**Important:** If your site uses clustering, changes to default settings for runtime-server properties must be applied to other server nodes manually. The settings in `run.properties` are not replicated using automated synchronization methods.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pf.admin.https.port</td>
<td>Defines the port on which the PingFederate administrative console runs. Default is 9999.</td>
</tr>
<tr>
<td>pf.console.bind.address</td>
<td>Defines the IP address over which the PingFederate administrative console communicates. Use for deployments where multiple network interfaces are installed on the machine running PingFederate.</td>
</tr>
<tr>
<td>pf.console.login.mode</td>
<td>Indicates whether more than one Admin user may access the administrative console at one time (see “Setting Administration Options” on page 58). Values: Single</td>
</tr>
<tr>
<td>pf.console.authentication</td>
<td>Indicates whether administrators log on to PingFederate using credentials managed internally, by PingFederate, or externally (see “Alternative Console Authentication” on page 38).</td>
</tr>
<tr>
<td>ldap.properties.file</td>
<td>When LDAP administrative-console authentication is enabled, indicates the name of the file containing configuration properties.</td>
</tr>
<tr>
<td>pf.http.port</td>
<td>Defines the port on which PingFederate listens for unencrypted HTTP traffic at runtime. For security reasons, this port is turned off by default.</td>
</tr>
<tr>
<td></td>
<td><strong>Caution:</strong> This port should remain disabled in production if your deployment configuration directly exposes the PingFederate server to the Internet.</td>
</tr>
<tr>
<td>pf.https.port</td>
<td>Defines the port on which PingFederate listens for encrypted HTTPS (SSL/TLS) traffic. Default is 9031.</td>
</tr>
</tbody>
</table>
pf.secondary.https.port Defines a secondary HTTPS port, for use on the back channel, to facilitate interoperability with other federation software vendors. To use this port, change the placeholder value to the port number you want to use, if needed.

Additional configuration of the listener ports (including adding new listeners) is available via the `<pf_install>/pingfederate/server/default/deploy/jetty.sar/META-INF/jboss-service.xml` file. Of particular value are the `WantClientAuth` and `NeedClientAuth` flags, which indicate to a client the request or requirement, respectively, for a client certificate. (For this port, `WantClientAuth` is set to true by default; `NeedClientAuth` is set to false.)

pf.engine.bind.address Defines the IP address over which the PingFederate server communicates with partner federation gateways. Use for deployments where multiple network interfaces are installed on the machine running PingFederate.

pf.monitor.bind.address Defines the IP address over which an SNMP agent and JMX communicate with PingFederate (see "Configuring Runtime Reporting" on page 60). Use for deployments where multiple network interfaces are installed on the machine running PingFederate.

pf.hsm.mode Enables or disables (the default) a FIPS-compliance Hardware Security Module (see Appendix A in Getting Started).

pf.provisioner.mode Enables or disables (the default) SaaS Provisioning (see "SaaS Provisioning" on page 18). Also used to enable provisioning failover (see the PingFederate Server Clustering Guide).
Installing a New License Key

If your license key is going to expire or has expired recently, you can install a new one without interrupting services.

**Tip:** You can configure the server to send administrators email warnings regarding the license status (see “Configuring Runtime Notifications” on page 59).

You will also need to install a new license key as a new user, or after you obtain PingFederate software updates (other than patch releases).

**To request a new license key:**


After you receive the new license key, you can either install it manually or by using the PingFederate administrative console, with or without a previous license installed.

**Tip:** When you use the administrative console and a previous license exists, PingFederate compares the new license with the existing one and warns of any potential problems (or aborts the import if the license is invalid for any reason). After the license key is imported, the previous license file is saved with a timestamp in the filename.

**To install a license key manually:**

1. Copy the license file as `pingfederate.lic` file to:
   `<pf_install>/pingfederate/server/default/conf`

**Tip:** For manual installation, the license file *must* be named `pingfederate.lic`. It may take up to a minute for a running server to recognize a new key.

**To install a license key using the administrative console (when no previous license is configured):**

1. Start PingFederate and access the administrative console (see “Starting and Stopping PingFederate” on page 24).
2. On the Import License screen, click **Browse** to select the file and then click **Import**.
   
   The license file is verified for use with the current instance of PingFederate and copied as `pingfederate.lic` file to:
   
   `<pf_install>/pingfederate/server/default/conf`

**To install a replacement license key from the Main Menu:**

1. Click **License Management** under Administrative Functions.
2. On the License Management screen, click **Browse** to select the file and then click **Import**.

No filename restrictions are imposed for importing the file; it is renamed before installation if necessary. The previous `pingfederate.lic` license file is renamed with a timestamp in the configuration directory:

```
<pf_install>/pingfederate/server/default/conf
```

If the new license does not include support for features covered by your existing license, or if there is some other potential problem with the license, you will be advised and prompted on whether to continue.

**Note:** If the license is for the wrong version of PingFederate or is found to be invalid for some other reason, PingFederate displays the error(s) and reverts to the previous license.

### Automating Configuration Migration

PingFederate provides a configuration-migration tool that can be used for scripting the transfer of administrative-console configurations from one PingFederate server to another—for example, from a test environment to production. The command-line utility, `configcopy` in `<pf_install>/pingfederate/bin`, uses PingFederate's built-in Connection Management Web Service in conjunction with an internal Web Service to export and import configurations, and to obtain lists (see “Connection Management Service” on page 345).

**Important:** The Connection Management Service must be activated for both the source and target servers before the `configcopy` tool can be used (see “Application Authentication” on page 95.)

This tool performs these overall configuration-transfer processing steps:

1. Retrieves configuration data (XML) from a source PingFederate server.
2. Modifies the configuration with any changes required for the target environment, according to settings in one or more properties files and/or command-line arguments.
3. Imports the updated configuration into the PingFederate target server.

The `configcopy` tool can perform these functions in real time, from server to server, or by using an intermediate file. The latter option is useful when both the source and target PingFederate servers are either not running at the same time or not accessible from the same operating-system command window.

**Important:** For one-time configuration transfers from one version of PingFederate to a newer version, use a configuration archive rather than the `configcopy` tool (see “Using the Configuration Archive Utility” on page 32). The tool is not supported for use across versions.

General operational capabilities include:

- Listing of source partner connections, adapter or STS token-translator instances, SaaS-provisioning channels, or data-store connections.

List commands include optional filter settings, when applicable.
Automating Configuration Migration

- Copying one or more partner connections, SaaS-provisioning channels, or instances of adapters or token translators.
- Copying one or more data-store connections.
- Copying server settings.

Using the Migration Tool

The configuration-migration utility works in conjunction with one or more optional properties files, which may be used to define the operational command and other parameters, including the source and/or target PingFederate servers, and to modify configuration settings as needed for the target environment. A consolidated template for the properties file, `configcopy.conf`, is located with the tool itself in `<pf_install>/pingfederate/bin`.

**Note:** Refer to `configcopy.conf` for detailed information, including a list of command options and associated parameters, descriptions of all operational properties, and lists of all Override Properties (configuration settings that can be modified in transit).

Templates are also available for each command option in `<pf_install>/pingfederate/bin/configcopy_templates`. The templates break down the properties according to the applicable commands.

Copies of the templates can be configured as needed and then used together (or combined into one file). Use the applicable filenames as an argument when running `configcopy.bat` or `configcopy.sh` (depending on your operating system) for particular configurations, using the following command syntax:

(On Windows)
```
configcopy.bat
-Dconfigcopy.conf.file=<properties_file1>;<properties_file2>;...
```

When paths are included with the filenames, you cannot use backslashes (`\`). Use forward slashes (`/`) or escape the backslash (`\\`).

(On Unix/Linux)
```
configcopy.sh
-Dconfigcopy.conf.file=<properties_file1>:<properties_file2>:...
```

Note that the file separators are platform specific, corresponding to the syntax used for system-level path separators.

Alternatively (or in addition), you can specify any property values via command-execution arguments, using the following syntax:

```
configcopy[.sh] -D<property>=<value> ...
```

where `<property>` is any property named in the properties file and `<value>` is the value.
Chapter 2
System Administration

Command-line property designations take precedence over any values set in the properties file.

Note: Access to the Connection Management Web Services are password-protected (see "Application Authentication" on page 95). The usernames and passwords may be set in the properties file for both the source and target Web Services (passwords can be obfuscated). If passwords are set in the properties file, they cannot be overridden using the command line. If a password is not set, the configcopy tool prompts for it. Usernames always must be supplied where applicable, either in the command line or in the properties file.

The configcopy utility generates its own log file, configcopy.log, which is located in the <pf_install>/pingfederate/log directory. You can control settings for this log, as needed, in the file configcopy.log4j.properties, located in the bin directory.

Caution: The import process at the target server is not subject to the same rigorous data validation performed by the administrative console during manual configuration. Although some checks are made, it is possible to create invalid connections using the connection-migration process. Therefore, you should not use the configcopy tool to attempt to create settings at the target that do not exist at the source; the tool is designed only for modifying the values of existing source settings to make them applicable to the target environment.

In addition, to avoid errors and prevent unstable target configurations due to missing components or faulty cross-component references (for example, invalid ID references from connection configurations to data-store configurations), be sure to adhere closely to the instructions provided in the following procedure.

To use configcopy:

1. Ensure access to the Connection Management Web Service is enabled for both the source and target PingFederate servers (see "Application Authentication" on page 95).

2. Determine which component configurations need to be copied, including plugins.

   For example, connection configurations always reference either adapter or token-translator configurations (or both) and may reference JDBC or custom data-store configurations. These are all separate configurations, and must be copied separately (unless they already exist at the target) in conjunction with copying connection configurations.

   Server Settings, unless preconfigured at the target, also need to be copied over separately.

   SaaS Provisioning settings are part of connection configurations but may be copied separately as needed to update target connections.

3. Ensure necessary plug-in JAR files are installed on the target server.

   The configcopy tool does not copy over these files, which include libraries for adapters, token translators, and JDBC or any custom database drivers.

   The JAR files are located in the directory:
   <pf_install>/pingfederate/server/default/deploy
4. On the target server, ensure that signing certificates (or certificates used for XML decryption) are already in place (see “Security Management” on page 87). Private keys are not copied from server to server (public certificates may be copied). Make note of identifying information about the target keys so you can reference the certificates in connection-copy properties.

5. If you have not yet installed your organization’s (CA-issued) SSL server certificate on both the target and source servers, either do so or use one of the following work-arounds to ensure that configcopy can contact both servers:
   - (Recommended) Install the Issuer certificate for the PingFederate SSL certificate in a separately managed trust store. Then the location of the file can be specified when running configcopy using the property `configcopy.connection.trust.keystore`.
   - Install the Issuer certificate for the PingFederate SSL certificate into the trust store for the JDK under which configcopy runs.

   **Note:** If different SSL certificates are installed on the two servers, the configcopy tool must be able to trust both. In this case, both certificates must be installed in the trust store used by configcopy, or in the trust store for the JDK under which configcopy runs.

6. Create properties files for the necessary commands and associated command-parameter values needed to copy the required configurations. Refer to the README.txt file and to the properties-file templates in the directory: `<pf_install>/pingfederate/bin/configcopy_templates`

   **Note:** This step and those following assume the use of properties files based on the templates provided; you may also use command-line parameters (see information earlier in this section).

7. If you are copying connections, override ID properties referencing adapter, data stores or other plug-in configurations, as needed (see Step 2).

   **Important:** Ensure that the plug-in configurations are either previously defined at the target or are part of the same configcopy process used to copy the connections that depend on them.

8. Create a script or run a command (or command series) that executes configcopy for each of the prepared properties files. See the discussion above for syntax requirements, or the README file.

**SaaS Provisioning CLI**

PingFederate provides as command-line interface (CLI) to help manage automated provisioning for SaaS users at IdP sites (see “SaaS Provisioning” on page 18). Administrators can use this tool to view the status of user provisioning, either globally or one provisioning channel at a time, and to rectify unusual situations where provisioning at the SaaS provider may get out of sync with the enterprise user store (see “Configuring SaaS Provisioning” on page 170).
The CLI tool, provmgr.bat or provmgr.sh, is located in the directory <pf_install>/pingfederate/bin. The tool interacts with the internal data store PingFederate uses to maintain provisioning synchronization between the LDAP user store and the target service (see “Configuring SaaS Provisioning Settings” on page 69).

Note that the tool creates its own log file, provmgr.log, located in the directory <pf_install>/pingfederate/log. You can control settings for this log, as needed, in the file provmgr.log4j.properties, located in the bin directory.

The following tables describes the available global and channel-specific command arguments:

<table>
<thead>
<tr>
<th>Table 5: SaaS Provisioning CLI Global Options</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Command Argument</strong></td>
</tr>
<tr>
<td>---</td>
</tr>
<tr>
<td>--help</td>
</tr>
</tbody>
</table>
| --show-channels | Lists all channels in a table format, showing for each:
- ID - A numeric channel ID (channel-specific commands need this ID)
- Name - The channel name
- Connection ID
- Status (active/inactive) - Both the connection and the channel status are shown (see “Channel Activation and Summary” on page 181)
- User count/dirty-user-record count (e.g.: 5000/12 means 5000 users and 12 dirty records)
- Source (as LDAP URL)
- Target code |
| --show-nodes | Shows all the provisioning-server nodes with their status and the last timestamp (applies only to a failover configuration—see the PingFederate Server Cluster Guide in the installation docs directory). |
| --force-node-backup | Sets the provisioner mode to FAILOVER for the associated PingFederate server node (see the Server Clustering Guide). |

Note: With each command, specify the channel with the argument:
-c <channel-id-number>

Example:
```
provmgr -c 1 --show-source
```

You can determine channel ID numbers by using the global command:
```
provmgr --show-channels
```
<table>
<thead>
<tr>
<th>Command Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--reset-group-timestamp</td>
<td>Deletes the group timestamp, which forces the provisioner to process the provisioning group on the next cycle, even if the timestamp on that group did not actually change.</td>
</tr>
<tr>
<td></td>
<td>Depending on your LDAP server and administrative practices, you may want to schedule this command to run periodically to catch up with any users that may have been deleted (rather than deactivated) in the directory server: some directory servers do not update the group timestamp for deleted users.</td>
</tr>
<tr>
<td></td>
<td><strong>Important</strong>: This option should seldom be needed if users are deactivated rather than deleted. If it is needed, you may wish to schedule it when other network activity is low.</td>
</tr>
<tr>
<td>--reset-attribute-sync</td>
<td>Sets the attribute sync timestamp to 1, which forces the provisioner to look at all users for changes, not only those that have a newer timestamp on their LDAP entry.</td>
</tr>
<tr>
<td></td>
<td><strong>Important</strong>: This option should be needed rarely and may consume considerable network resources, depending on the number of users. If it is needed, you may wish to schedule it when other network activity is low.</td>
</tr>
<tr>
<td>--reset-values-hash</td>
<td>Removes the values hash for all users. (The database stores a hash of attribute values for users to determine whether any values have been changed.)</td>
</tr>
<tr>
<td></td>
<td>This argument forces users that have a newer timestamp on their LDAP entry to be updated at the SaaS provider, regardless of the actual field values. Note, however, that users whose recorded timestamp is unchanged are <em>not</em> updated.</td>
</tr>
<tr>
<td>--reset-all</td>
<td>Equivalent to using all three of the arguments above.</td>
</tr>
<tr>
<td></td>
<td><strong>Important</strong>: This option should be needed rarely if ever and may consume considerable network resources, depending on the number of users. If it is needed, you may wish to schedule it when other network activity is low.</td>
</tr>
<tr>
<td>--show-dirty-records</td>
<td>Lists all users who have not been provisioned or updated at the SaaS site. Each entry shows:</td>
</tr>
<tr>
<td></td>
<td>• LDAP GUID</td>
</tr>
<tr>
<td></td>
<td>• SaaS Username</td>
</tr>
<tr>
<td></td>
<td>• SaaS GUID</td>
</tr>
</tbody>
</table>
Customizing User-Facing Screens

PingFederate supplies HTML templates to provide information to the end user or to request user input during SSO/SLO processing. These template pages utilize the Velocity template engine, an open-source Apache project, and are located in the `<pf_install>/pingfederate/server/default/conf/template` directory.

You can modify most of these pages in a text editor to suit the particular branding and informational needs of your PingFederate installation. Each page contains both Velocity constructs and standard HTML. The Velocity engine interprets the

<table>
<thead>
<tr>
<th>Command Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--show-user</td>
<td>Shows all internal database fields related to the specified user, including</td>
</tr>
<tr>
<td></td>
<td>transitory mapping fields (fields waiting to be pushed to the SaaS provider);</td>
</tr>
<tr>
<td></td>
<td>also shows all LDAP attributes retrieved from the directory server for this</td>
</tr>
<tr>
<td></td>
<td>user.</td>
</tr>
<tr>
<td>Note:</td>
<td>You can obtain usernames and GUIDs for dirty user records, as needed, using</td>
</tr>
<tr>
<td></td>
<td>the --show-dirty-records option.</td>
</tr>
<tr>
<td></td>
<td>The LDAP GUID, if used and if it is binary, should be entered in hexadecimal</td>
</tr>
<tr>
<td></td>
<td>format (as shown in log files).</td>
</tr>
<tr>
<td>Examples:</td>
<td>provmgr.sh --show-user -u john@example,com</td>
</tr>
<tr>
<td></td>
<td>provmgr.sh --show-user -g ffd448643f812b43a0bee2504173f0</td>
</tr>
<tr>
<td>--clear-dirty-records</td>
<td>Clears the dirty flag on all user records.</td>
</tr>
<tr>
<td>--delete-dirty-records</td>
<td>Removes all dirty user records from the internal store.</td>
</tr>
<tr>
<td>--delete-all-users</td>
<td>Removes all users from the internal store and deletes the provisioning group</td>
</tr>
<tr>
<td></td>
<td>timestamp and the last attribute-sync timestamp.</td>
</tr>
<tr>
<td></td>
<td>The effect of this command is to reset the channel to its initial state. All</td>
</tr>
<tr>
<td></td>
<td>user metadata is lost and provisioning for the channel will start from the</td>
</tr>
<tr>
<td></td>
<td>beginning, picking up all users and pushing them to the SaaS provider when</td>
</tr>
<tr>
<td></td>
<td>the synchronization frequency interval is expired (see “Configuring SaaS</td>
</tr>
<tr>
<td></td>
<td>Provisioning Settings” on page 69).</td>
</tr>
<tr>
<td>Important:</td>
<td>This option should be needed rarely if ever. If it is needed, you may wish</td>
</tr>
<tr>
<td></td>
<td>to schedule it when other network activity is low.</td>
</tr>
<tr>
<td>--show-source</td>
<td>Displays all source LDAP configuration parameters, including settings and</td>
</tr>
<tr>
<td></td>
<td>location.</td>
</tr>
<tr>
<td>--show-target</td>
<td>Displays the target configuration.</td>
</tr>
</tbody>
</table>

PingFederate supplies HTML templates to provide information to the end user or to request user input during SSO/SLO processing. These template pages utilize the Velocity template engine, an open-source Apache project, and are located in the `<pf_install>/pingfederate/server/default/conf/template` directory.

You can modify most of these pages in a text editor to suit the particular branding and informational needs of your PingFederate installation. Each page contains both Velocity constructs and standard HTML. The Velocity engine interprets the
Customizing User-Facing Screens

commands embedded in the template page before the HTML is rendered in the user's browser. At runtime, the PingFederate server supplies values for the Velocity variables used in the template.

**Tip:** Variables indicated in each file are the only variables that can be used for rendering the associated Web-browser page.

For information about Velocity, please refer to the Velocity project documentation on the Apache Web site: [http://velocity.apache.org/engine/releases/velocity-1.4](http://velocity.apache.org/engine/releases/velocity-1.4)

Changing Velocity or Javascript code is not recommended.

At runtime, the user's browser is directed to the appropriate page, depending on the operation being performed and where the related condition occurs (see tables below). For example, if an SSO error occurs during IdP-initiated SSO, the user's browser is directed to the IdP's SSO error-handling page.

Applications can override the PingFederate server-hosted pages provided specifically for SSO and SLO errors by specifying a URL value in the relevant endpoint’s `InErrorResource` parameter (see “Application Endpoints” on page 335). Administrators can override SSO/SLO success pages by specifying default URLs in the administrative console (for the IdP configuration, see “Configuring a Default URL and Error Message” on page 107; for the SP, see “Configuring Default URLs” on page 200).

The following tables describe each of the templates. To help identify the templates from the end user's point of view, the tables are organized by the titles that appear at the top of the user's browser window.

**Note:** Because the templates can be customized, the default titles listed in the tables may not be accurate for your PingFederate installation or your partner’s.

<table>
<thead>
<tr>
<th>Page Title</th>
<th>Purpose</th>
<th>Type</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Error - Single Logout</strong></td>
<td>Displayed when an SLO request fails and no other SLO error landing page is specified.</td>
<td>Error</td>
<td>User should close browser</td>
</tr>
<tr>
<td><code>idp.slo.error.page.template.html</code></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Error - Single Sign-On</strong></td>
<td>Displayed when IdP-initiated SSO fails and no other SSO error landing page is specified. Displays system errors and information for the user.</td>
<td>Error</td>
<td>Consult log and Web developer</td>
</tr>
<tr>
<td><code>idp.sso.error.page.template.html</code></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Please Select Authentication System</strong></td>
<td>Displayed when the user must choose from several IdP security domains. Based on the user's selection, the server redirects the browser to the appropriate adapter instance for authentication.</td>
<td>Normal</td>
<td>User must make selection</td>
</tr>
<tr>
<td><code>sourceid-choose-idp-adapter-form-template.html</code></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Signed Out</strong></td>
<td>Indicates user signed out of the IdP under the WS-Federation protocol and lists each successful SP logout, when applicable.</td>
<td>Normal</td>
<td>None</td>
</tr>
<tr>
<td><code>sourceid-wsfed-idp-signout-cleanup-template.html</code></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 7: IdP User-Facing Pages (Continued)

<table>
<thead>
<tr>
<th>Page Title and template file name</th>
<th>Purpose</th>
<th>Type</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Signing Out</td>
<td>WS-Federation IdP sign-out processing page. <strong>Note</strong>: No HTML is rendered in the browser.</td>
<td>Normal</td>
<td>None</td>
</tr>
<tr>
<td>Success - Single Logout</td>
<td>Displayed when an SLO request succeeds but no other SLO landing page is specified.</td>
<td>Normal</td>
<td>None</td>
</tr>
<tr>
<td>Working . . .</td>
<td>Used to auto-submit a WS-Federation assertion to the SP. If Javascript is disabled, the user is prompted to click a button to POST the assertion directly. <strong>Note</strong>: Normally not displayed if Javascript executes properly.</td>
<td>Normal</td>
<td>None</td>
</tr>
</tbody>
</table>

Table 8: SP User-Facing Pages

<table>
<thead>
<tr>
<th>Page Title and template file name</th>
<th>Purpose</th>
<th>Type</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Account Link Removed</td>
<td>Communicates a user’s successful “defederation” operation.</td>
<td>Normal</td>
<td>None</td>
</tr>
<tr>
<td>Account Linking</td>
<td>Used to authenticate a user at the SP when an account link needs to be established.</td>
<td>Normal</td>
<td>None</td>
</tr>
<tr>
<td>Authentication Failed</td>
<td>Displayed when an authentication challenge fails during WS-Federation processing.</td>
<td>Error</td>
<td>Consult log and Web developer</td>
</tr>
<tr>
<td>Error - Single Logout</td>
<td>Displayed when an SLO request fails and no other SLO error landing page is specified.</td>
<td>Error</td>
<td>User should close the browser</td>
</tr>
<tr>
<td>Error - Single Sign-On</td>
<td>Displayed when SP-initiated SSO fails and no other SSO error landing page is specified.</td>
<td>Error</td>
<td>Consult log and Web developer</td>
</tr>
<tr>
<td>Select Identity Provider</td>
<td>The user requested SP-initiated SSO, but the IdP partner was not specified in the appropriate query parameter or cookie. This page allows the user to select the IdP manually. Based on the user’s selection, the server redirects the browser to the appropriate IdP partner’s SSO service.</td>
<td>Normal</td>
<td>User must make selection</td>
</tr>
<tr>
<td>Signed Out - Service Provider</td>
<td>Displays the user’s sign-out status.</td>
<td>Normal</td>
<td>None</td>
</tr>
</tbody>
</table>
### Table 8: SP User-Facing Pages (Continued)

<table>
<thead>
<tr>
<th>Page Title and template file name</th>
<th>Purpose</th>
<th>Type</th>
<th>Action</th>
</tr>
</thead>
</table>
| **Success - Single Logout**  
  sp.slo.success.page.template.html | Displayed when an SLO request succeeds and no other SLO success landing page is specified. | Normal | None |
| **Single Sign-On Target Unspecified**  
  sp.sso.success.page.template.html | Displayed when an SSO request succeeds but no target-resource parameter is specified by the incoming URL, and no default URL is set (see “Configuring Default URLs” on page 200). | Error | Consult Web developer, or specify default URL |

### Table 9: Either IdP or SP User-Facing Pages

<table>
<thead>
<tr>
<th>Page Title and template file name</th>
<th>Purpose</th>
<th>Type</th>
<th>Action</th>
</tr>
</thead>
</table>
| **Error - Single Sign-On**  
  generic.error.msg.page.template.html | For an Auto-Connect SSO transaction, indicates a range of possible error conditions (see “Using Auto-Connect” on page 15):  
  - The requesting Auto-Connect partner is not found in the PingFederate server’s list of allowed domains.  
  - The partner’s metadata is not accessible.  
  - The server is not configured for Auto-Connect.  
  - General error, with error code. | Error | Consult log, check configuration, or contact partner. If unresolved, contact Ping Identity support. |
| **Error**  
  general.error.page.template.html | Indicates that an unknown error has occurred and provides a error reference number and (optionally) an error message. | Error | Consult log, contact Ping Identity support |
| **Sign On**  
  AbstractPasswordIdpAuthnAdapter.form.template.html | Challenges user for credentials when authentication can take place via HTTP Basic Authentication or an HTML form, depending on the operational mode. | Normal | User must sign on |
| **Submit Form**  
  form.autopost.template.html | Whenever the server posts a form, this template is used to auto-submit the form. If Javascript is disabled, the user is prompted to click a button to post the form manually.  
  **Note:** Normally not displayed if Javascript executes properly. | Normal | None |
Chapter 3

System Settings

The System Settings links on the Main Menu (under My Server) provide access to global settings that may apply to either an IdP or an SP federation configuration.

This chapter covers:

- “Managing Server Settings” on page 57
- “Managing Data Stores” on page 72
- “Configuring IdP Discovery” on page 81

Note: The information in this chapter is presented from the viewpoint of an administrative user with “Admin” permissions (see “Account Management” on page 34).

Managing Server Settings

Server settings include unique federation server identifiers, the designation of your site’s federation role (SP, IdP, or both), and your enabled federation protocols (see the “Supported Standards” chapter in Getting Started).

Server settings also include system-administration configuration (one-user or multi-user), email notification options and setup, and a shortcut link to account management (when multi-user administration is enabled).

If you have enabled Auto-Connect and/or a SaaS Connector, then you will need to configure several parameters specific to those features in the System Settings task flow.

You configure many of these settings initially during the installation setup (see “Running PingFederate for the First Time” in the “Installation” chapter of Getting Started), but you can change or add to them as needed from the Main Menu.

Note: For information about WS-Trust STS Settings, see “Configuring STS Authentication” on page 271).
Information in this section covers:

- “Setting Administration Options” on page 58
- “Entering System Information” on page 59
- “Configuring Runtime Notifications” on page 59
- “Configuring Runtime Reporting” on page 60
- “Managing Accounts” on page 63
- “Choosing Roles and Protocols” on page 64
- “Specifying Federation Information” on page 67
- “Configuring SaaS Provisioning Settings” on page 69
- “Configuring Auto-Connect Metadata Signing” on page 70
- “Configuring Auto-Connect Metadata Lifetime” on page 71
- “Saving and Editing Server Settings” on page 72

Setting Administration Options

On the System Administration screen, PingFederate provides a choice of single- or multi-user access to the administrative console.

**Note:** If you are using your network’s LDAP user-data store or client certificates for administrative-console authentication, this screen is not presented (see “Alternative Console Authentication” on page 38).

<table>
<thead>
<tr>
<th>Configuring My Server</th>
<th>Help</th>
<th>Support</th>
<th>About</th>
<th>Logout (Administrator)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Main</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>System Administration</strong></td>
<td><strong>System Info</strong></td>
<td><strong>Runtime Notifications</strong></td>
<td><strong>Runtime Reporting</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Account Management</strong></td>
<td><strong>Roles &amp; Protocols</strong></td>
<td><strong>Federation Info</strong></td>
<td><strong>WS-Trust STS Settings</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Summary</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

If you choose single-user administration, the console is accessible only by using the default Administrator ID, for which full privileges are provided. If you choose multi-user administration, the system provides role-based access control (see “Account Management” on page 34).

**Note:** To return to single-user administration after having previously enabled multi-user, you must have only one user marked as active under Account Management.

Also on this screen you can choose to use different SSL server certificates for the administrative console and for PingFederate runtime processing. To fully implement this choice, you must import two (or more) server certificates into PingFederate (see “SSL Server Certificates” on page 88).
To reach this screen:
1. Click Server Settings on the Main Menu.
2. Click System Administration under the Server Settings tab.

Entering System Information

On the System Info screen, you provide general information about your company.

To reach this screen:
1. Click Server Settings on the Main Menu.
2. Click System Info under the Server Settings tab.

Configuring Runtime Notifications

Depending on your licensing agreement, your PingFederate license may have an expiration date. Under Runtime Notifications you can set up the server to send an email warning when your license is about to expire.

**Note:** The license-notification option does not appear if you have a perpetual license.

You can also configure the server to send an email notification to a specific administrator (or a group) when a certificate used by PingFederate is about to expire, or has expired.

**Note:** When a certificate expires, PingFederate always writes an error in the server log, regardless of whether runtime notification is configured (see "Log File Generation" on page 24).
To reach this screen:
1. Click Server Settings on the Main Menu.
2. Click Runtime Notifications on the Summary screen.

To configure notifications:
1. Click the checkbox next to the type of notification you want, and enter an email address.
2. If you are configuring certificate-expiration notification, enter an advance-warning time period in the Initial Warning field (optional) and in the Final Warning field.

   **Note:** When advance certificate-expiration notification is configured, the server also sends notification if a license expires.

3. If you have not previously configured PingFederate to access your email server, click Email Server Settings (see “Managing Email Configuration” on page 40).

**Configuring Runtime Reporting**

PingFederate supports runtime monitoring and reporting through the Simple Network Management Protocol (SNMP), a standard used by network-management consoles to monitor network and server activity across an enterprise.

In addition, PingFederate supports runtime monitoring and reporting through Java Management Extensions (JMX) (see “Runtime Monitoring Using JMX” on page 61).

**Using SNMP Monitoring**

The SNMP Management Information Base (MIB) is included in the docs/SNMP directory of the PingFederate distribution. The MIB describes the object identifiers that PingFederate uses to communicate information through SNMP.
identifiers are globally unique and managed by the Internet Assigned Numbers Authority (IANA).

Configure details of SNMP monitoring, if your site uses this form of monitoring, on the Runtime Reporting screen.

SNMP supports Guards and Traps. A Guard is a request for status information sent by a network-management console to an SNMP agent. Embedded within each PingFederate server is an SNMP agent that brokers the communication between the management console and PingFederate.

PingFederate responds to two types of Guard requests:

- The total number of transactions the server has processed since installation
- The total number of failed transactions that the server has encountered since installation

A Trap is a spontaneous communication from an agent to a network-management console. PingFederate generates a Trap at regular intervals—the server “heartbeat.” Each Trap contains the amount of time the server instance has been running since its most recent start-up.

- If you configure Traps, change settings as needed and then click Test SNMP Configuration to send a single Trap to your network-management console.

You can also use an HTTP call at any time to verify that the PingFederate server is running (see “Maintenance Endpoint” on page 343).

**Runtime Monitoring Using JMX**

Similar to SNMP, JMX technology represents a Java-centric approach to application management and monitoring. JMX exposes instrumented code in the form of Managed Beans (MBeans). Application management systems that support JMX
technology—for example, the standard Sun JDK client, JConsole—may request runtime information from PingFederate’s JMX server.

PingFederate’s JMX server reports monitoring data for SSO and SLO transactions as well as for SaaS Provisioning (see “SaaS Provisioning” on page 18). In addition, because PingFederate is built within an existing JBoss framework, numerous JBoss-standard MBeans are available to the PingFederate server’s JMX clients.

**Important:** Authentication is required for JMX-client access to PingFederate runtime data (see “Application Authentication” on page 95).

### SSO/SLO Monitoring

For SSO/SLO transaction processing, PingFederate provides these MBeans:

- `pingfederate:type=TOTAL_FAILED_TRANSACTIONS`
- `pingfederate:type=TOTAL_TRANSACTIONS`

Each type contains a single attribute, `Count`, which reports the same information as an SNMP `Get` (see “Configuring Runtime Reporting” on page 60).

**Note:** Counts are reset when the PingFederate server is restarted.

### Provisioning Monitoring

For SaaS Provisioning, PingFederate provides an MBean called `pf.provisioning:type=saas.provisioning.events`. The MBean exposes five JMX Operations, each corresponding to the Java methods described in Table 10. Each method returns a `CompositeData` object, which allows for the retrieval of complex data without requiring application-specific code to reside with the JMX client.

**Table 10 SaaS Provisioning JMX Monitoring Options**

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>viewEvents</code></td>
<td>Gets an array of specific events based on the given criteria. The parameters filter the data collectively; that is, they are joined logically by “and”.</td>
<td><code>wassSuccessful</code> – If true, returns information only on successful transactions; <code>false</code> returns information only on failed transactions; <code>null</code> returns all transactions.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>eventTypeStr</code> – The type of event. Valid values are: CREATE, UPDATE, DISABLE, ENABLE; <code>null</code> or an empty string returns all types.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>fromDate</code> – See Note below.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>toDate</code> – See Note below.</td>
</tr>
</tbody>
</table>

**Important:** Authentication is required for JMX-client access to PingFederate runtime data (see “Application Authentication” on page 95).
Managing Server Settings

### Advanced JMX Configuration

By default, PingFederate uses port 1099 for its JMX server. To change the port or other JMS configuration items, if needed, modify the configuration file `jmx-remote-config.xml` in the directory `<pf_install>/server/default/data/config-store`.

**Note:** When connecting to the JMX service using SSL (the default), ensure that the client trusts the PingFederate SSL server certificate presented (see “SSL Server Certificates” on page 88). (This should be a consideration only during testing, when using the certificate installed with PingFederate or another self-signed certificate.)

### Managing Accounts

When you choose multi-user system administration, you can create users during installation or while configuring Server Settings (see “Setting Administration Options” on page 58).
Alternatively, you can set up and maintain user accounts later as a separate task (assuming you have user administration permissions—see “Account Management” on page 34). By default for installation, the user “Administrator” has full system permissions.

Note: If you are using your network’s LDAP user-data store or client certificates for PingFederate authentication, the Account Management screen is not presented (see “Alternative Console Authentication” on page 38).

To continue, click Next or Save.

For information about adding or managing users, see “Account Management” on page 34.

Choosing Roles and Protocols

On the Roles and Protocols screen, select which identity-federation role(s) your organization plays and which sets of standards you will use with your federation partner (see the “Supported Standards” chapter in Getting Started).

Note: If you are using the PingFederate WS-Trust STS, select WS-Trust as one of the supported protocols for either an IdP, an SP, or both. Notice that a new configuration step, WS-Trust STS Settings, appears under the Server Settings tab. For information about this configuration, see the “Server Settings” section under “WS-Trust STS Configuration” on page 269.
Also on this screen, you can choose any of several options:

- As an IdP, if you have installed a PingFederate SaaS Connector, you can enable the SaaS Provisioning option (see “SaaS Provisioning” on page 18).
- As an SP, if you are using SAML 2.0 XASP for multiple IdP connections, you may choose to have PingFederate determine dynamically which connection to use (see “Attribute Query and XASP” in the “Supported Standards” chapter of Getting Started).
- For either role you can enable Auto-Connect for SAML 2.0 connections (see “Using Auto-Connect” on page 15).
- If you are installing PingFederate and are not sure of your selections, just click Next.

**Note:** If you do not choose a role during installation, you must return to this screen to do so before you can configure connections to federation partners.

To reach this screen for editing:

1. On the Main Menu under System Settings, click **Server Settings**.
2. Click **Roles and Protocols** under the Server Settings tab.
To choose roles and protocols:

1. Select your federation role(s) and then select at least one protocol.

   **Note:** SaaS Provisioning requires the use of the SAML 2.0. (For more information, refer to the Quick Connection Guide contained in the PingFederate SaaS Connector package for your service provider.)

2. (Optional) If you are using SAML 2.0 and want to configure Auto-Connect, select that feature for your role(s) (see “Using Auto-Connect” on page 15).

   **Note:** Clearing this checkbox does not deactivate an existing Auto-Connect configuration in production. If you have already deployed Auto-Connect and wish to suspend the deployment for any reason, use the Initial Setup Summary screens (accessible from the Main Menu) for your respective role.

   When you make this selection, two additional steps are added to the System Settings task:
   - Metadata Signing (see “Configuring Auto-Connect Metadata Signing” on page 70)
   - Metadata Lifetime (see “Configuring Auto-Connect Metadata Lifetime” on page 71).

3. (Optional) If you are using PingFederate as an IdP and have installed a SaaS Connector package, select the SaaS Provisioning checkbox.

   (For more information, see “SaaS Provisioning” on page 18.)

   **Note:** After provisioning is configured for a connection, you cannot clear this checkbox—you must delete all provisioning configurations first. To suspend provisioning for an SP partner, you can deactivate the specific configuration (see “Channel Activation and Summary” on page 181). Alternatively, you can deactivate the associated SP connection; note, however, that this will also disable SSO/SLO transactions (see “Connection Activation and Summary” on page 182).

4. (Optional) If you are using SAML 2.0 XASP as an SP for multiple IdP connections, you may select the option to determine dynamically which connection to use, based on the X.509 certificate presented (see “Attribute Requester Mapping” on page 203).

   **Tip:** After you make this selection and create XASP IdP connections (see “Configuring the Attribute Query Profile” on page 157), configure dynamic IdP discovery via the Attribute Requester Mapping link on the Main Menu. Once the mapping is configured, you cannot clear the checkbox on the Roles and Protocols screen unless you first delete the mapping.

   For general information about XASP, see “Attribute Query and XASP” in the “Supported Standards” chapter of Getting Started.

5. Click Next (or Save, if you are modifying existing selections).
For information about configuring settings associated with your selections, see these relevant portions of this manual:

- Chapter 5, “Identity Provider SSO Configuration”
- Chapter 6, “Service Provider SSO Configuration”
- Chapter 7, “WS-Trust STS Configuration”
- “Configuring IdP Discovery” on page 81

### Specifying Federation Information

This information identifies your federation deployment to your partners, according to the protocol(s) you support.

**Notes:** You must provide an ID that uniquely identifies your federation gateway for each protocol you support. For WS-Trust STS, IDs are required for both SAML 2.0 and SAML 1.x, regardless of browser-based SSO protocol support or the type of token expected to be issued, to ensure that the STS will perform correctly under all conditions.

Each ID normally applies across all connection partners for a given protocol; however, if your implementation requires different IDs for the same protocol, you can use virtual server IDs (see “Federation Server Identification” on page 20).

You can also use a different ID for Auto-Connect transactions (see “Using Auto-Connect” on page 15).
### Field Descriptions

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Base URL</strong></td>
<td>The fully qualified host name, port, and path (if applicable) on which the PingFederate server runs. This field is used to populate configuration settings in metadata files (see “Exporting Metadata” on page 27).</td>
</tr>
<tr>
<td><strong>SAML 2.0 Entity ID</strong></td>
<td>This ID defines your organization as the entity operating the server for SAML 2.0 transactions. It is usually defined as an organization’s URL or a DNS address; for example: pingidentity.com. The SAML SourceID used for artifact resolution is derived from this ID using SHA1.</td>
</tr>
<tr>
<td><strong>Auto-Connect Entity ID</strong></td>
<td>(Optional) If you are using Auto-Connect, you can specify a unique ID here for Auto-Connect processing. The value must be a fully qualified URL and should match the CN of your Auto-Connect certificates (see “Auto-Connect Security Model” on page 17). When a value is supplied, this ID is used instead of the SAML 2.0 Entity ID in your server’s Auto-Connect metadata, as well as in associated SSO/SLO requests and responses. Use this field if you have configured regular, static SAML 2.0 connections to other partners and your SAML 2.0 Entity ID is not a fully qualified URL (see “Using Auto-Connect” on page 15).</td>
</tr>
</tbody>
</table>
Managing Server Settings

To reach this screen:
1. Click Server Settings on the Main Menu.
2. Click Federation Info under the Server Settings tab.

### Configuring SaaS Provisioning Settings

On the SaaS Provisioning screen, you can select the database that PingFederate uses internally to facilitate provisioning for SaaS providers when PingFederate is configured as an IdP (see “SaaS Provisioning” on page 18).

This screen is presented only if you have installed a SaaS Connector package and SaaS Provisioning is enabled for the IdP federation role (see “Choosing Roles and Protocols” on page 64).

⚠️ **Caution:** A pre-installed, default Hypersonic database is selected for initial setup and testing. However, we strongly recommend that you choose your own, secured database for production deployments.

On this screen, you can also change the provisioning synchronization frequency—that is, how often PingFederate checks the local user store for changes.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAML 1.x Issuer/Audience</td>
<td>This ID identifies your federation server for SAML 1.x transactions. As with SAML 2.0, it is usually defined as an organization’s URL or a DNS address. The SourceID used for artifact resolution is derived from this ID using SHA1.</td>
</tr>
<tr>
<td>SAML 1.x Source ID</td>
<td>(Optional) If supplied, the Source ID value entered here is used for SAML 1.x, instead of being derived from the SAML 1.x Issuer/Audience.</td>
</tr>
<tr>
<td>WS-Federation Realm</td>
<td>The URI of the realm associated with the PingFederate server. A realm represents a single unit of security administration or trust.</td>
</tr>
</tbody>
</table>

**Note:** The fields available on this screen depend on the federation protocols enabled on your server (see “Choosing Roles and Protocols” on page 64).
The database is used to store the state of synchronization between the source data store and the target data store, enabling periodic checking to determine whether updates are required at the target SaaS site. (For information on configuring provisioning as an IdP, see “Configuring SaaS Provisioning” on page 170.)

**Note:** PingFederate has been tested using Hypersonic, Oracle, and MySQL databases as internal provisioning data stores. However, any relational database should work as well; adaptable setup scripts used for Hypersonic, Oracle, and MySQL are provided in the directory: `<pf_install>/pingfederate/server/default/conf/provisioner/sql-scripts`

To configure the internal data store:

1. Select the data store from the drop-down list.
   
   If the data store you want is not shown in the list, then PingFederate is not yet configured to access the store; click **Manage Data Stores** to create a connection to the data store (see “Managing Data Stores” on page 72).

2. (Optional) Change the Synchronization Frequency value.

### Configuring Auto-Connect Metadata Signing

When Auto-Connect is enabled, PingFederate generates publicly available, signed metadata for partners to use. The metadata contains information about your server configuration (see “Providing Metadata” on page 15).

On the Metadata Signing screen, you choose a certificate to use for signing the metadata.

**Important:** The certificate CN must match the domain name associated with the Entity ID configured in the previous screen (see “Specifying Federation Information” on page 67).
This screen appears only if Auto-Connect is enabled for either an IdP or SP (see “Choosing Roles and Protocols” on page 64).

If you have not yet imported the certificate you want, click Manage Certificates (see “Security Management” on page 87).

Note: The signing certificate is included as part of the public metadata and must be trusted by your partner (see “Auto-Connect Security Model” on page 17).

Configuring Auto-Connect Metadata Lifetime

Partners using Auto-Connect metadata will cache it to use for the future requests during the “lifetime” in which the metadata is valid, as configured on the Metadata Lifetime screen. After the metadata lifetime is expired, the metadata is retrieved again.

This metadata expiration ensures that partners always have reasonably up-to-date information about your server. You may elect to use the default time period or change it on the Metadata Lifetime screen.

This screen appears only if Auto-Connect is enabled for either an IdP or SP (see “Choosing Roles and Protocols” on page 64).
Saving and Editing Server Settings

On the Server Settings Summary screen you can view, edit, and save your configuration.

- Click **Save** if you are finished with this configuration, or click any heading to make changes.

Managing Data Stores

PingFederate can connect to local data stores to retrieve user attributes on either the IdP or SP side of an SSO transaction (or both).

**Tip:** Whenever attributes are retrieved from a data store at runtime, PingFederate logs the activity (see "Log File Generation" on page 24). When you set up access to a data store, you can choose to mask the values of all retrieved attributes in the log files to enhance security and privacy of personal information (see "Attribute Masking" on page 9).

As an IdP, you use this feature whenever you need to fulfill an attribute contract that requires information beyond that which can be derived from the user's session (see “Configuring Attribute Sources and User Lookup” on page 133). For example, this information may include such attributes as an email address, a job title, or any data that can be used to customize a user's experience at the SP site.

As an SP, you can use data stores to retrieve additional attributes to package with the IdP’s assertion data to meet SP adapter requirements (see “SSO Integration Kits and Adapters” on page 4). Such attributes may be needed, for example, to establish authorization levels or to manage the local account.

Either IdP or SP organizations configuring PingFederate for user provisioning must set up connections to data stores (see “User Provisioning” on page 17).

You can add data stores at any time. Standard data stores include JDBC-enabled databases and LDAP v3-compliant directories. Alternatively, you can develop a driver using the PingFederate Custom Source SDK to connect to non-JDBC databases (see the PingFederate SDK Developer’s Guide in the pingfederate/sdk directory).

**Note:** You cannot delete or modify a data-store connection if it is associated with an attribute source as part of a partner-connection configuration. You must remove the association first.

<table>
<thead>
<tr>
<th>Configuring My Server</th>
<th>Help</th>
<th>Support</th>
<th>About</th>
<th>Logout (Administrator)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Manage Data Stores</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

To reach this screen:

- Click **Data Stores** on the Main Menu.
To add a data store:
1. Click Add New Data Store.
2. Select Database, LDAP, or Custom and click Next.
3. Continue the configuration:
   • For Database configuration information see “Configuring a JDBC Database Connection” on page 73.
   • For LDAP configuration information see “Configuring an LDAP Connection” on page 77.
   • For Custom configuration information see “Configuring a Custom Data Store” on page 79.
4. Click Save when you return to this screen.

   \[\text{Important: Note this step; be sure to click Save.}\]

To modify a data store:
1. Click the data store Description.
   • For Database configuration information see “Configuring a JDBC Database Connection” on page 73.
   • For LDAP configuration information see “Configuring an LDAP Connection” on page 77.
   • For Custom configuration information see “Configuring a Custom Data Store” on page 79.

   \[\text{Important: You must have current connectivity from PingFederate to a data store in order to create or modify the configuration. If you find that the configuration is not editable, then your connection has been lost due to a system problem not related to the PingFederate server. The problem must be identified and corrected before you can continue.}\]

To delete a data store:
1. Click delete under Action for the data store you want to delete.
   (To undo the deletion, click undelete.)
2. Click Save.

Configuring a JDBC Database Connection
You configure access to a database by providing basic JDBC information.

\[\text{Note: Ensure that your database driver JAR file is installed in the pingfederate/server/default/lib directory. You must restart the server after installing the driver.}\]
Field Descriptions

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JDBC URL</td>
<td>The location of the JDBC database. For example, <code>jdbc:mysql://databaseservername/databasename</code> where <code>databaseservername</code> is the DNS host name (or IP) of the server hosting the database, and <code>databasename</code> is the name of a database on that server.</td>
</tr>
<tr>
<td>Driver Class</td>
<td>The name of the driver class used to communicate with the source database. For example, <code>org.hsqldb.jdbcdriver</code>. This class should be supplied by the database software vendor in a JAR file, which must be present in the <code>pingfederate/server/default/lib</code> directory.</td>
</tr>
<tr>
<td>Username</td>
<td>The name that identifies the user when connecting to the database.</td>
</tr>
<tr>
<td>Password</td>
<td>The password needed to access the database.</td>
</tr>
<tr>
<td>Mask Values in Log</td>
<td>(Optional) Determines whether all attribute values returned from this data store will be masked in PingFederate log files (see &quot;Attribute Masking&quot; on page 9).</td>
</tr>
<tr>
<td>Validate Connection SQL</td>
<td>(Optional, but recommended) An SQL statement used by JBoss at runtime to validate that the connection is still active. Refer to your JDBC or JBoss documentation for more information.</td>
</tr>
</tbody>
</table>

To reach this screen:

1. Click **Data Stores** on the Main Menu.
2. Click the Data Store Description link on the Manage Data Stores screen.

**To configure a new data store:**
1. Click Data Stores on the Main Menu.
2. Click Add New Data Store.
3. Select Database and click Next.

**To establish access to a database:**
1. Enter the applicable JDBC URL.
   This URL is used to identify the data store in lists. Example:
   `jdbc:mysql://10.0.1.81:3306/idp`
2. Enter the Driver Class.
   Example: `com.mysql.jdbc.Driver`
3. Enter a valid Username and Password.
4. (Optional) Select Mask Values in Log.
   For information see “Attribute Masking” on page 9.
5. (Optional) Click Advanced.
   Use this option to change default sizes or look-up time-outs, or to validate the connection using a specific SQL call (see “Setting Advanced Options” on page 75).
6. Click Next.

**Note:** The driver JAR file must be loaded into the directory:
```
pingfederate/server/default/lib
```
7. On the Summary screen, click Done.
8. Click Save on the Manage Data Stores screen.

**Setting Advanced Options**

Use the Advanced Database Options screen to change default pool sizes or look-up time-outs, or to validate the connection using a specific SQL call.
Chapter 3
System Settings

Field Descriptions

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum Pool Size</td>
<td>The smallest number of database connections in the connection pool for the given data store.</td>
</tr>
<tr>
<td>Maximum Pool Size</td>
<td>The largest number of database connections in the connection pool for the given data store.</td>
</tr>
<tr>
<td>Blocking Timeout (ms)</td>
<td>The amount of time a request waits to get a connection from the connection pool before it fails.</td>
</tr>
<tr>
<td>Idle Timeout (ms)</td>
<td>The length of time the connection can be idle in the pool before it is closed.</td>
</tr>
</tbody>
</table>

To reach this screen for editing:
1. Click Data Stores on the Main Menu.
2. Click the Data Store Description link on the Manage Data Stores screen.
3. Click the Advanced button on the Database Config screen.

To configure a new data store:
1. Click Data Stores on the Main Menu.
2. Click Add New Data Store.
3. Select Database and click Next.
4. Enter information on the Database Config screen and click the Advanced button.
   Internally, PingFederate is preconfigured to use published JBoss server default values. To view or restore these values, click Apply Defaults.
Configuring an LDAP Connection

This screen establishes a connection between the PingFederate server and an LDAP data store.

Field Descriptions

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hostname(s)</td>
<td>The DNS name or IP address of the data store, which may include a port number; example: 181.20.42.130:389. For failover, you can enter one or more backup LDAP servers, each separated by a space. Note: If more than one Hostname is entered, each server must be accessible using the same User DN and Password (or via Bind Anonymously).</td>
</tr>
<tr>
<td>LDAP Type</td>
<td>If you are using this data store for SaaS Provisioning and your LDAP store is either Active Directory or Sun Directory Server, select the applicable Type (see “SaaS Provisioning” on page 18). Identifying the LDAP type permits PingFederate to configure many provisioning settings automatically (see “Modifying Source Settings” on page 174). Tip: If you are using SaaS Provisioning and your LDAP server is not Active Directory or Sun, you may wish to define a custom LDAP Type to streamline the provisioning configuration (see “Defining an LDAP Type” on page 79). If you are not using SaaS Provisioning, the selection of an LDAP Type has no effect.</td>
</tr>
<tr>
<td>Bind Anonymously</td>
<td>(Optional) Username and password are not required (see procedure below).</td>
</tr>
</tbody>
</table>
To establish a connection to an LDAP data store:

1. Enter the applicable Hostname(s).
   For more information, see the description for this field in the table above.
   Hostnames identify this LDAP configuration in selection lists elsewhere in the 
   administrative console.

2. (Optional) Select an LDAP Type from the drop-down list.
   For more information, see the description for this selection in the table above.

3. Either:
   - Check Bind Anonymously if your LDAP interface supports anonymous 
     binding and if no credentials are needed to access the data store.
   Or:
   - Enter a valid User DN and Password.

   **Note:** If you choose an anonymous binding, ensure that this access 
   level provides permission to search the directory for user-account 
   information.

4. (Optional) Select Use SSL.
   For more information, see the description for this selection in the table above.

5. (Optional) Select Mask Values in Log.
   For more information, see the description for this selection in the table above.

6. Click **Next**.
   PingFederate will try to connect to the LDAP server at this point. If it cannot, 
   there may be a problem with your settings.

7. On the Summary screen, click **Done**.

8. Click **Save** on the Manage Data Stores screen.
Defining an LDAP Type

If you are using SaaS Provisioning and your user-management LDAP server is not Active Directory or the Sun Directory Server, you can define a custom LDAP Type for PingFederate to use to streamline the SaaS provisioning configuration (see “SaaS Provisioning” on page 18).

When the LDAP server is defined, its type appears in the LDAP Type drop-down list on the LDAP Configuration screen (see previous section). When the data store is selected as the source for provisioning, a number of other settings can be automatically configured (see “Modifying Source Settings” on page 174).

To define an LDAP Type:

1. If you are using the LDAP Configuration screen, click Previous or Cancel.
2. Copy and rename the file sample.template.txt:
   `<pf_install>/pingfederate/server/default/conf/template/ldap-templates`
3. Change the template.name in the new template file.
   The template.name you specify will appear in the LDAP Type list on the LDAP Configuration screen when you save the template.
4. Modify other property values in the file to match the corresponding configuration of your LDAP server.
   The properties are used in the SaaS Provisioning setup (see “Modifying Source Settings” on page 174).
5. Save the new template file.

Setting Pooling Options

LDAP connection pooling is maintained by the Java runtime environment. Normally, default settings are optimal. However, if you need to customize pooling for particular applications, you can find a list of the properties controlling pooling at:


Using the format `<prop_name>=<value>`, enter any of the properties into the run.properties file located in the directory:

   `<pf_install>/pingfederate/bin`

Configuring a Custom Data Store

Developers can use the PingFederate Custom Source SDK to create specific drivers for non-JDBC/LDAP data stores (or more sophisticated JDBC/LDAP lookups) including, for example, flat files or SOAP-connected databases (see the PingFederate SDK Developer’s Guide in the pingfederate/sdk directory).

Once the data-store driver is installed, you can select it on the Custom Data Store Type page.
To start configuring a Custom Data Store:

1. Enter a unique Instance Name.
   You can create more than one instance of the same Data Store Type for use with different connection partners, as needed.

2. Select the Data Store Type.

3. (Optional) Select Mask Values in Log.
   For information see “Attribute Masking” on page 9.

4. Click Next.

Configuring a Custom Data Store Instance

This screen will vary depending on the implementation. Below is a sample for a SOAP-enabled database driver. The screen shown below is only an example of a custom data store and is not available in the PingFederate distribution.

To configure the driver instance for use with a partner connection:

- Enter or select required information and click Next.
Invoking Adapter Actions

Custom data store adapters may be written to interface PingFederate to perform configuration assistance or validation actions (for example, testing a connection to a database). Actions may also include generation of parameters that might need to be set manually in a configuration file.

- To invoke an adapter action (when applicable), click its link on the Adapter Actions screen.

Editing and Saving a Data Store

On the Data Store Summary page, you can view or edit your configuration.

To modify the configuration:
- Click the heading above the information you want to change.

To save a new configuration:
- Click Done on the Summary screen and then Save on the Manage Data Stores screen.

Configuring IdP Discovery

PingFederate provides two kinds of IdP discovery:

- SAML 2.0 standard IdP Discovery (see “IdP Discovery” in the “Supported Standards” chapter of Getting Started)
- Proprietary IdP discovery using a persistent cookie written by an SP PingFederate server

Standard IdP Discovery is configured in the administrative console (see the next section).

Discovery based on a PingFederate proprietary cookie is configured in an XML file (see “IdP Discovery Using a Persistent Cookie” on page 84). Note that this method can be used in conjunction with any of the federation standards.

Standard IdP Discovery

SAML IdP Discovery provides a cookie-based look-up mechanism used to identify a user's IdP dynamically during an SP-initiated SSO event, when the IdP is not otherwise specified. To enable this feature, IdP Discovery must be selected on the Roles and Protocols screen in the System Settings configuration (see “Choosing Roles and Protocols” on page 64). Then click IdP Discovery under System Settings on the Main Menu to reach this screen:
To continue, click Configure IdP Discovery.

Choosing Domain Cookie Settings

On the Domain Cookie Settings screen, you choose the discovery role or roles that PingFederate will play.

The choices that appear on this screen depend on whether PingFederate is acting as an SP, an IdP, or both; or as an IdP Discovery server only (see “Choosing Roles and Protocols” on page 64).

To reach this screen:

1. Click IdP Discovery under System Settings on the Main Menu. If this link is not available, then IdP Discovery is not yet enabled (see “Choosing Roles and Protocols” on page 64).

2. On the IdP Discovery screen, click Configure IdP Discovery.

For a detailed discussion of selections on this screen, see “IdP Discovery” in the “Supported Standards” chapter of Getting Started.

Configuring a Common Domain Service

A Common Domain Service is where PingFederate reads and/or writes authentication information contained in shared cookies, as determined by whether
your site is an SP or IdP, respectively. (The service is shared if your PingFederate server is acting in both roles.)

To reach this screen:
1. Click **IdP Discovery** under **System Settings** on the **Main Menu**.
   
   If this link is not available, then IdP Discovery is not yet enabled (see “Choosing Roles and Protocols” on page 64).
2. On the IdP Discovery screen, click **Configure IdP Discovery**.
3. Click **Common Domain Service** under the Configure IdP Discovery tab.
   
   This step is not available if your server is configured for IdP Discovery only (see “Choosing Roles and Protocols” on page 64).

To configure the Common Domain Service:
1. Enter the **Base URL**.
   
   You must use SSL/TLS (HTTPS) for a common domain.
2. Enter and confirm a **Pass phrase** that a Web application must use to access the domain.
Configuring a Local Common Domain Server

A Local Common Domain Server is where PingFederate reads (as an SP) or writes (as an IdP) cookies for IdP Discovery.

To reach this screen:

1. Click **IdP Discovery** under System Settings on the Main Menu.
   
   If this link is not available, then IdP Discovery is not yet enabled (see “Choosing Roles and Protocols” on page 64).

2. On the **IdP Discovery** screen, click **Configure IdP Discovery**.

3. Click **Local Common Domain Server** under the **Configure IdP Discovery** tab.

   This step is available only if the common-server option is selected under Domain Cookie Settings (see “Configuring IdP Discovery” on page 81).

To configure the Local Common Domain Server:

1. Enter the **Common Domain**.
   
   Your entry must include an initial period (.), for example: `.pingidentity.com`

2. Enter the **Cookie Lifetime (days)**.
   
   The range is 1 to 1,825 days; or to indicate a nonpersistent, session cookie, enter -1.

3. Enter and confirm a **Pass phrase** that a Web application must use to access the domain.

Editing and Saving the Configuration

After configuring or modifying IdP Discovery settings, you can review the configuration on the Summary screen.

- If you are finished with the configuration, click **Save**; otherwise, click any heading to make changes.

IdP Discovery Using a Persistent Cookie

PingFederate's proprietary IdP-discovery method makes use of an IdP Persistent Reference Cookie (IPRC) to track the identity provider with whom a user last
Configuring IdP Discovery

Administrator's Manual 85

authenticated. There are three significant differences between standard IdP Discovery and the IPRC method:

- Standard IdP Discovery may be used only with SAML 2.0; the IPRC may be used with any federation protocol.
- The Common Domain Cookie (CDC) may be configured as a temporary, session-based cookie; the IPRC always persists for a configurable period of time.
- The CDC is set by the IdP and readable by both federation partners; the IPRC is set by the SP, using information in the SAML assertion, and cannot be accessed by the IdP.

Configuration

Enable the IPRC feature for your SP site using the configuration file org.sourceid.websso.profiles.sp.IdpIdCookieSupport.xml located in the directory <pf_install>/pingfederate/server/default/data/config-store.

Note that the deployed connection configuration between SP and IdP partners must include SP-initiated SSO (see “Configuring Protocol Settings” on page 234).

To enable IPRC:

1. In the XML configuration file cited above, set the value of EnableIdpIdCookie to true.

2. (Optional) Change the default value(s) of any of the remaining elements in the configuration, as described in the following table:

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IdpIdCookieName</td>
<td>The name of the IPRC set by the SP installation (default: IdPId). Note that the cookie name cannot contain any of the following characters: &gt; , &lt; comma, semicolon, space.</td>
</tr>
<tr>
<td>IdpIdCookieLifeTimeInDays</td>
<td>The maximum lifetime for the cookie (default 365 days). The browser will delete the cookie when the period is expired.</td>
</tr>
<tr>
<td>ShowIdpSelectionList</td>
<td>If set to true (the default), the SP displays a list of IdPs that can be used to initiate the SSO event if the cookie is not set. If set to false, the SP installation generates an error page.</td>
</tr>
</tbody>
</table>

3. Start or restart PingFederate.

Note: Once an IPRC cookie is set, the only way to change the IdP to whom the SP will send Authentication Requests for the user is to do one of the following: wait for the cookie to expire, delete the cookie, or perform IdP-initiated SSO using the new IdP.
PingFederate provides built-in certificate management to handle security considerations for SAML transactions. In addition, when you use the SAML 2.0 Attribute Query profile as an SP, password security is required between the application requesting attributes and the SP PingFederate server. Basic authentication is also required for applications making calls to PingFederate’s Connection Management Service and optional for the SSO Directory Service (see “Web Service Interfaces” on page 345).

**Note:** PingFederate does not support Certificate Extensions.

You manage certificates and application authentication via the Security section under My Server on the Main Menu.

### Security

- Trusted CAs
- SSL Server Certificates
- SSL Client Keys & Certificates
- Digital Signing & XML Encryption Keys & Certificates
- Application Authentication
- Certificate Revocation Checking

**Note:** The information in this chapter is presented from the viewpoint of an administrative user with “Crypto Admin” permissions (see “Account Management” on page 34).

## Trusted Certificate Authorities

You can import your federation partner’s CA-signed or self-signed SSL/TLS server certificate(s) into PingFederate’s global trust list. If the CA is not one of the major
authorities, you may also need to import the certificate from the Certificate Authority that signed the certificates into the global trust list.

**Note:** If a required CA certificate is already available in cacerts in the Java runtime, it is not necessary to import the same certificate into the PingFederate store.

**To reach this screen:**

- Click **Trusted CAs** on the Main Menu.

**To import a certificate:**

1. Click **Import**.
2. Click **Browse** to locate the certificate.
3. Highlight the file and click **Open**.
4. Click **Next**.
5. Click **Done**.
6. Click **Save** on the Manage Trusted CAs screen.

**To export a certificate:**

1. Click **Export** under Action for the certificate you want to export.
2. On the Summary page, click the **Export** button.
3. Save the file on your system.

**To delete a certificate:**

1. Click **Delete** under Action for the certificate you want to delete.
   - To undo the deletion, click **Undelete**.
2. Click **Save**.

**To view certificate details:**

- Click the certificate Serial number.

---

**SSL Server Certificates**

PingFederate provides built-in SSL/TLS certificate management. Use this feature to establish and maintain the certificate(s) presented for access to the PingFederate administrative console and for incoming SSL/TLS connections at runtime (see “Setting Administration Options” on page 58).
To create a new certificate:
1. Click Create New.
2. Enter the requested information on the form.
3. Click Next.
4. On the Summary screen, click Done.
5. Click Save on the Manage SSL Server Certificates screen.

To import a certificate and private key:
1. Click Import.
2. Click Browse to locate the certificate.
3. Highlight the file and click Open.
4. Enter the certificate password.
5. Click Next.
6. Click Done.
7. Click Save on the Manage SSL Server Certificates screen.

To view certificate information:
- Click its Serial number.

Note: If a certificate has been revoked, PingFederate indicates this problem in the certificate information window.

To activate a certificate:
1. Click Activate for Runtime Server or Activate for Admin Console under Action for the certificate you want to activate.
   These choices are enabled only if you have created or imported more than one certificate. Otherwise, a single certificate is used for both the administrative console and runtime operations.
2. Click Save on the Manage SSL Server Certificates screen.

To export a certificate:
1. Click Export under Action for the certificate you want to export.
2. Select **Certificate Only** on the Export Certificate screen.
   Or:
   Select **Certificate and Private Key** and enter an Encryption Password.
3. Click **Next**.
4. On the Certificate Summary screen, click **Export**.
5. Save the file on your system and click **Done**.

**To create a certificate-authority signing request:**
1. Click **Certificate Signing** under **Action** for the desired certificate.

   ![Note: This selection is inactive if you have not yet saved a newly created or imported certificate. Click Save and then return to this screen from the Main Menu.]

   2. Select Generate Certificate Signing Request (CSR), if not already selected.
3. Click **Next**.
4. Click **Generate CSR** on the Generate CSR screen.
5. Click **Next**.
6. On the Certificate Summary screen, click **Export**.
7. Save the file on your system and click **Done**.

**To import a certificate authority response:**
1. Click **Certificate Signing** under **Action** for the relevant certificate.
2. Select Import CSR Response and click **Next**.
3. Click **Browse** and locate the CSR response to import.
4. Highlight the file and click **Open**.
5. Click **Next**.
6. Click **Done** on the Summary screen.
7. Click **Save** on the Manage SSL Server Certificates screen.

**To delete a certificate:**
1. Click **Delete** under **Action** for the certificate you want to delete.
   To undo the deletion, click **Undelete**.
2. Click **Save**.

---

**Create Certificate Field Descriptions**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Common Name</td>
<td>The common name (CN) identifying the certificate.</td>
</tr>
<tr>
<td>Organization</td>
<td>The organization (O) or company name creating the certificate.</td>
</tr>
<tr>
<td>Organizational Unit</td>
<td>The specific unit within the organization (OU).</td>
</tr>
<tr>
<td>City</td>
<td>The city or other primary location (L) where the company operates.</td>
</tr>
</tbody>
</table>
SSL Client Keys & Certificates

You can create and manage your authentication private keys and the certificates your server presents as a client in an SSL/TLS transaction.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>State</td>
<td>The state (ST) or other political unit encompassing the location.</td>
</tr>
<tr>
<td>Country</td>
<td>The country (C) where the company is based.</td>
</tr>
<tr>
<td>Validity (days)</td>
<td>The time during which the certificate is valid.</td>
</tr>
<tr>
<td>Key Algorithm (drop-down menu)</td>
<td>A mathematical formula used to generate a key. PingFederate uses either of</td>
</tr>
<tr>
<td>Key Size (bits)</td>
<td>The number of bits used in the key. (RSA-768 to 2048, DSA-768 and 1024)</td>
</tr>
</tbody>
</table>

SSL Client Keys & Certificates

You can create and manage your authentication private keys and the certificates your server presents as a client in an SSL/TLS transaction.

To reach this screen:

- Click SSL Client Keys & Certificates on the Main Menu.

To create a new certificate:

1. Click Create New.
2. Enter the information on the form.
3. Click Next.
4. On the Summary screen, click Done.
5. Click Save on the Manage SSL Auth Private Keys and Certificates screen.

To import a certificate:

1. Click Import.
2. Click Browse to locate the certificate.
3. Highlight the file and click Open.
4. Enter the certificate password.
5. Click Next.
6. Click Done on the Import Certificate Details screen.
7. Click Save on the Manage SSL Auth Private Keys and Certificates screen.
To view certificate information:
- Click the certificate Serial number.

Note: If a certificate has been revoked, PingFederate indicates this problem in the certificate information window.

To export a certificate:
1. Click Export under Action for the certificate you want to export.
2. Select Certificate Only.
   Or:
   Select Certificate and Private Key and enter an Encryption Password.
3. Click Next.
5. Save the file on your system and click Done.

To create a certificate-authority signing request:
1. Click Certificate Signing under Action for the desired certificate.
2. Select Generate Certificate Signing Request (CSR), if not already selected.
3. Click Next.
4. Click Generate CSR on the Generate CSR screen.
5. Click Next.
7. Save the file on your system and click Done.

To import a certificate authority response:
1. Click Certificate Signing under Action for the relevant certificate.
2. Select Import CSR Response and click Next.
3. Click Browse and locate the CSR response to import.
4. Highlight the file and click Open.
5. Click Next.
6. Click Done on the Summary screen.
7. Click Save on the Manage SSL Auth Private Keys and Certificates screen.

To delete a certificate:
1. Click Delete under Action for the certificate you want to delete.
   (To undo the deletion, click Undelete.)
2. Click Save.
Create Certificate Field Descriptions

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Common Name</td>
<td>The common name (CN) identifying the certificate.</td>
</tr>
<tr>
<td>Organization</td>
<td>The organization (O) or company name creating the certificate.</td>
</tr>
<tr>
<td>Organizational Unit</td>
<td>The specific unit within the organization (OU).</td>
</tr>
<tr>
<td>City</td>
<td>The city or other primary location (L) where the company operates.</td>
</tr>
<tr>
<td>State</td>
<td>The state (ST) or other political unit encompassing the location.</td>
</tr>
<tr>
<td>Country</td>
<td>The country (C) where the company is based.</td>
</tr>
<tr>
<td>Validity (days)</td>
<td>The time during which the certificate is valid.</td>
</tr>
<tr>
<td>Key Algorithm (drop-down menu)</td>
<td>A mathematical formula used to generate a key. PingFederate uses either of two algorithms, RSA or DSA.</td>
</tr>
<tr>
<td>Key Size (bits)</td>
<td>The number of bits used in the key. (RSA-768 to 2048, DSA-768 and 1024)</td>
</tr>
</tbody>
</table>

Digital Signing and Decryption Keys & Certificates

You can use PingFederate to create and maintain your server's signing certificates, which you may use to sign SAML requests, responses, and assertions. You can also use these certificates for XML decryption (“XML Encryption” on page 14).

To reach this screen:

- Click Digital Signing & XML Decryption Keys & Certificates on the Main Menu.
To create a new certificate:
1. Click Create New.
2. Enter the information on the form.
3. Click Next.
4. On the Summary screen, click Done.
5. Click Save.

To import a certificate:
1. Click Import.
2. Click Browse to locate the certificate.
3. Highlight the file and click Open.
4. Enter the certificate password.
5. Click Next.
6. Click Done.
7. Click Save.

To view certificate information:
- Click the certificate Serial number.

Note: If a certificate has been revoked, PingFederate indicates this problem in the certificate information window.

To export a certificate:
1. Click Export under Action for the certificate you want to export.
2. Select Certificate Only.
   Or:
   Select Certificate and Private Key and enter an Encryption Password.
3. Click Next.
5. Save the file on your system and click Done.

To create a certificate-authority signing request:
1. Click Certificate Signing under Action for the desired certificate.

Note: This selection is inactive if you have not yet saved a newly created or imported certificate. Click Save and then return to this screen from the Main Menu.

2. Select Generate Certificate Signing Request (CSR), if not already selected.
3. Click Next.
4. Click Generate CSR on the Generate CSR screen.
5. Click Next.
7. Save the file on your system and click Done.
To import a certificate authority response:
1. Click **Certificate Signing** under Action for the relevant certificate.
2. Select Import CSR Response and click **Next**.
3. Click **Browse** and locate the CSR response to import.
4. Highlight the file and click **Open**.
5. Click **Next**.
6. Click **Done** on the Summary screen.
7. Click **Save** on the Manage Digital Signing Certificates screen.

To delete a certificate:
1. Click **Delete** under Action for the certificate you want to delete.
   (To undo the deletion, click **Undelete**.)
2. Click **Save**.

Create Certificate Field Descriptions

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Common Name</td>
<td>The common name (CN) identifying the certificate.</td>
</tr>
<tr>
<td>Organization</td>
<td>The organization (O) or company name creating the certificate.</td>
</tr>
<tr>
<td>Organizational Unit</td>
<td>The specific unit within the organization (OU).</td>
</tr>
<tr>
<td>City</td>
<td>The city or other primary location (L) where the company operates.</td>
</tr>
<tr>
<td>State</td>
<td>The state (ST) or other political unit encompassing the location.</td>
</tr>
<tr>
<td>Country</td>
<td>The country (C) where the company is based.</td>
</tr>
<tr>
<td>Validity (days)</td>
<td>The time during which the certificate is valid.</td>
</tr>
<tr>
<td>Key Algorithm (drop-down menu)</td>
<td>A mathematical formula used to generate a key. PingFederate uses either of two algorithms, RSA or DSA.</td>
</tr>
<tr>
<td>Key Size (bits)</td>
<td>The number of bits used in the key. (RSA-768 to 2048, DSA-768 and 1024)</td>
</tr>
</tbody>
</table>

Application Authentication

If you are using the SAML 2.0 Attribute Query profile as an SP, then the requesting application(s) at your site must authenticate to the PingFederate server (see “Attribute Query and XASP” in the “Supported Standards” chapter of Getting Started and “/sp/startAttributeQuery.ping” on page 342).

In addition, authentication is required to access PingFederate runtime data via JMX (see “Runtime Monitoring Using JMX” on page 61) or to make SOAP calls to the
Connection Management Web Service. Authentication is optional for the SSO Directory Service (see “Web Service Interfaces” on page 345).

**Note:** To help ensure network security, access to all of these services is deactivated when PingFederate is first installed.

Administrators can activate and configure authentication for any or all of the services on the Application Authentication screen.

**To reach this screen:**

- On the Main Menu, click **Application Authentication** under Security in the System Settings section.

**To enable access to a service:**

1. Click **Activate** for the Service under Action.
2. Where required, enter an Id, Shared Secret, and Confirm Shared Secret for the service.

   You and the application developer must agree to these values.

   This step is optional for the SSO Directory Service; the Service can be active without requiring authentication (the default setting).
3. Repeat the steps above for other Services, as needed.
4. Click **Save**.

**To change an application ID or password:**

- Replace the existing information in the necessary field(s) and click **Save**.
To block access to an active service:

- Click **Deactivate** for the Service under **Action** and then click **Save**.

**Tip:** Although not accessible when deactivated, the Connection Management Service and the SSO Directory Service are still deployed by default as part of PingFederate. If your organization does not plan to use one (or either) of these services, you may wish to remove the corresponding WAR file(s) from the `<pf_install>/pingfederate/server/deploy` or (and) `../deploy2` directories, respectively:

- `pf-ws.war`
- `pf-mgmt-ws.war`

---

**Certificate Revocation Checking**

By default at runtime, PingFederate attempts to retrieve a CRL to verify that a signing certificate has not been revoked, whenever a CRL distribution-point URL is included within the certificate (see "Certificate Validation" on page 11). Optionally, on the Manage Certificate Revocation screen you can enable and configure OCSP checking as the preferred verification method, depending on your requirements (see "OCSP Revocation Checking" on page 11).

OCSP can be used in place of CRL checking, or CRLs can be retained as a backup method (for failover).

**Note:** When OCSP is enabled, CRL checking is not done independently—only as a failover option for one or more OCSP failure conditions.

Also on the Manage Certificate Revocation screen, you can change system-default settings for CRL checking, as needed.
No configuration changes are necessary on this screen if OCSP is not required for your federation deployment and the CRL defaults are acceptable.

**To reach this screen:**


**Field Descriptions (For OCSP Checking)**

<table>
<thead>
<tr>
<th>Field/Selection</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable OCSP</td>
<td>Turns on OCSP certificate-revocation checking.</td>
</tr>
</tbody>
</table>
### Field/Selection

<table>
<thead>
<tr>
<th>Field/Selection</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default OCSP Responder URL</td>
<td>The location of a URL to use for certificate-revocation checking, a backup used only if the OCSP Responder URL is not contained in the certificate.</td>
</tr>
<tr>
<td>Default OCSP Responder Signature Verification Certificate</td>
<td>Certificate used to verify that the returned certificate status was sent from the Default OCSP Responder—required if the certificate is not included in the response (click Manage Certificates to import the verification certificate, as needed).</td>
</tr>
<tr>
<td>Do NOT allow Responder to use cached responses</td>
<td>When unchecked (the default), the OCSP Responder uses cached responses when available for the subject certificate (for an indicated period of time—see the description for “Next Update Grace Period,” below). If checked, PingFederate sends a nonce in the request to the Responder, effectively requiring that the status of the certificate be determined in real time. This option is intended to enhance the prevention of Internet replay attacks (in addition to timestamping), where required. <strong>Important</strong>: Making this selection may slow down OCSP response time for a request and will increase general processing overhead at the Responder site.</td>
</tr>
<tr>
<td>This Update Grace Period</td>
<td>For the response to be considered valid, the PingFederate server-clock time must correspond to the <code>&lt;thisUpdate&gt;</code> timestamp in the OCSP response, plus or minus the number of minutes set for this field (to compensate for clock variances).</td>
</tr>
<tr>
<td>Next Update Grace Period</td>
<td>If the response includes a <code>&lt;nextUpdate&gt;</code> timestamp indicating when updated certificate statuses will be available, then PingFederate checks to ensure that the timestamp is not earlier than the current server time, adding this grace period to compensate for clock variances.</td>
</tr>
<tr>
<td>Responder Timeout</td>
<td>The allowable response time before the OCSP Responder URL is considered unavailable and processing continues (see “OCSP Responder is Unavailable,” below).</td>
</tr>
<tr>
<td>Certificate is Unknown</td>
<td>The certificate does not fall under the purview of the CA associated with the OCSP Responder. The drop-down choices indicate whether an unknown certificate is to be considered valid or not, or whether to try CRL checking.</td>
</tr>
<tr>
<td>OCSP Responder is Unavailable</td>
<td>Indicates what action to take if the Responder cannot be reached.</td>
</tr>
<tr>
<td>OCSP Responder Returns Error</td>
<td>Indicates what action to take if the Responder returns an error.</td>
</tr>
</tbody>
</table>
### Field Descriptions (For CRL Checking)

<table>
<thead>
<tr>
<th>Field/Selection</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable CRL Checking</td>
<td>Enables CRL revocation checking (the default). <strong>Note</strong>: CRL checking must remain enabled if any selections for OCSP Error Handling include failover. If OCSP is enabled and no CRL failover is specified, then this selection has no effect.</td>
</tr>
<tr>
<td>Treat Unretrievable CRLs as Revoked</td>
<td>If checked, PingFederate immediately aborts the processing associated with the certificate. If unchecked, the server treats the certificate as valid but continues trying to retrieve the CRL.</td>
</tr>
<tr>
<td>Next Retry on Resolution Failure</td>
<td>Specifies the number of minutes the server waits before trying to retrieve a CRL when the previous attempt failed—applies only when the selection above (Treat Unretrievable CRLs as Revoked) is unchecked.</td>
</tr>
<tr>
<td>Next Retry on Next Update Expiration</td>
<td>How long the server waits before requesting a new CRL when the most recently retrieved CRL (in cache) has a next-update time in the past.</td>
</tr>
<tr>
<td>Verify CRL Signature</td>
<td>When checked (recommended), PingFederate verifies the CRL signature using the public key of the issuer, which must be in the certificate chain or in the list of Trusted CAs (see “Trusted Certificate Authorities” on page 87).</td>
</tr>
<tr>
<td>Proxy Settings</td>
<td>If CRL checking is routed through a proxy server, specify the server’s Host (DNS name or IP address) and the Port number. The same proxy information applies to OCSP checking, when enabled.</td>
</tr>
</tbody>
</table>
Identity Provider
SSO Configuration

In an IdP role, you use the PingFederate administrative console to configure local application-integration information and to manage connections to your SP-partner sites. You must configure Server Settings from the Main Menu to establish your site as an IdP before configuring connections to SPs (see “Choosing Roles and Protocols” on page 64).

Note that only one connection is needed per partner, even if you are targeting more than one Web application at the destination SP site. You can configure more than one connection, however, if your partner supports multiple protocols, or supports multiple federation IDs for the same protocol (see “Federation Server Identification” on page 20).

Under some conditions, you can enable SSO for an unlimited number of partners at once by configuring a single, common connection (see “Using Auto-Connect” on page 15). You can also expedite single end-to-end connections by taking advantage of PingFederate Express (see “About PingFederate Express” on page 14).

This chapter covers the following topics:

- “Application Integration Settings” on page 102
- “Viewing Protocol Endpoints” on page 108
- “Managing SP Connections” on page 110
- “Defining SP Affiliations” on page 183
- “Using the Express Connection Template” on page 186
- “Configuring SP Auto-Connect” on page 189

Note: This chapter applies to configuration settings needed for secure Internet SSO (“Browser SSO”). While there is some cross-over information also applicable to WS-Trust STS, if you are using PingFederate exclusively as an STS, start with “WS-Trust STS Configuration” on page 269.
Chapter 5
Identity Provider SSO Configuration

Application Integration Settings

The integration of local applications with PingFederate is the essential “first-mile” configuration that allows end-users to access protected resources across domains. This process is facilitated through the use of application-integration kits and a robust Software Development Kit (see “SSO Integration Kits and Adapters” on page 4).

Under Application Integration Settings on the Main Menu, you configure the IdP Adapters that PingFederate needs to interact with applications or access-management systems used to authenticate users at your site. You can also set a Default URL to which users may be directed during SLO, and you can look up system endpoints that application developers at your site need to access PingFederate’s SSO/SLO services.

**Note:** If your PingFederate configuration enables the WS-Trust STS, the selections under Application Integration Settings also include links for configuring plug-in **Token Processors** and, optionally, **STS Request Parameters**. Locate configuration information under “IdP Configuration for STS” on page 275.

Configuring IdP Adapters

An IdP adapter is used to look up session information and provide user identification to PingFederate (see “SSO Integration Kits and Adapters” on page 4).

You must configure at least one instance of an IdP adapter in order to set up connections to SP partners.

**Note:** If you are configuring either the OpenToken or the LDAP Adapter, see “Configuring the IdP OpenToken Adapter” on page 320 or “Configuring the IdP LDAP Adapter” on page 329, respectively.

<table>
<thead>
<tr>
<th>Configuring My Server</th>
<th>Help</th>
<th>Support</th>
<th>About</th>
<th>Logout (Administrator)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Main</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Manage IdP Adapter Instances</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

PingFederate uses adapters to authenticate users to your partners’ applications. Here you can manage “instances” of adapters that SP connections may use to fulfill Attribute Contracts sent to partners.

<table>
<thead>
<tr>
<th>Instance Name</th>
<th>Instance Id</th>
<th>Type</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>IdP Adapter</td>
<td>idadapter</td>
<td>IdP Sample Adapter</td>
<td>None Available - In Use</td>
</tr>
</tbody>
</table>

- You reach this screen by clicking Adapters under Application Integration Settings in My IdP Configuration.

**To configure a new instance:**
- Click **Create New Instance**.

**To edit an existing adapter instance:**
- Click the Instance Name and click the step you need to change.
To delete an adapter instance:
1. Click Delete next to the Instance Name. (To undo the deletion, click Undelete.)

   Note: This option is available only if the adapter instance is not in use for a connection.

2. Click Save to confirm the deletion.

Selecting an IdP Adapter Type

The first step in creating an adapter instance is choosing an adapter type. Available adapter types are determined by JAR files loaded in the `<pf_install>/pingfederate/server/default/deploy` directory. Some adapters are bundled with PingFederate (see “SSO Integration Kits and Adapters” on page 4). Other adapters and integration kits are available from the Ping Identity Web site (www.pingidentity.com).

### Field Descriptions

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instance Name</td>
<td>A descriptive name for the adapter instance—for example, an identity management system name.</td>
</tr>
<tr>
<td>Instance ID</td>
<td>An internal identifier for the adapter instance. Must be alphanumeric with no spaces.</td>
</tr>
<tr>
<td>Type</td>
<td>A list of deployed IdP adapter types available for creating an adapter instance for the server. A developer typically deploys any new adapter types before an administrator sets up a connection partner.</td>
</tr>
</tbody>
</table>

To define an adapter instance:
1. Enter the Instance Name and Instance Id on the Type screen.
2. Select the Type from the drop-down menu.

   If the adapter you need is not listed, click [Visit PingIdentity.com for additional types](https://www.pingidentity.com) to see if a suitable adapter is available from the PingFederate download site, or create your own adapter (see “SSO Integration Kits and Adapters” on page 4).
3. Click **Next** and enter information on subsequent screens for this adapter setup.

   **Tip:** The setup steps and information needed at those steps vary with the adapters deployed on your server (see “SSO Integration Kits and Adapters” on page 4). For information about configuring the adapters packaged with PingFederate, see “OpenToken Adapter Configuration” on page 319 or “LDAP Adapter Configuration” on page 327.

4. Click **Done** on the Adapter Summary screen.

5. Click **Save** on the Manage IdP Adapter Instances screen.

### Configuring an IdP Adapter

Depending on the adapter you choose, different configuration parameters are available on the IdP Adapter screen. These options are controlled by the adapter software (see “SSO Integration Kits and Adapters” on page 4).

- For information about configuring the OpenToken Adapter, see “OpenToken Adapter Configuration” on page 319.
- For information about configuring the LDAP Authentication Service, see “LDAP Adapter Configuration” on page 327.

**Important:** If you change adapters that are used by existing partner connections, you may need to reconfigure those connections. If so, a **Fix Errors** link appears on the Manage IdP Adapter Instances screen. Click the link to navigate to the screens you need to reconfigure. You cannot save the changes to the adapter until the existing connections have been repaired.

### Invoking Adapter Actions

Adapters may be written to provide configuration assistance or validation actions (for example, testing a connection to LDAP). Actions may also include generation of parameters that might need to be set manually in a configuration file.

- For information about actions available using the OpenToken Adapter, see “Configuring the IdP OpenToken Adapter” on page 320.
- For information about actions available using the LDAP Authentication Service, see “Configuring the IdP LDAP Adapter” on page 329.
To reach this screen:
1. Click Adapters on the Main Menu.
2. Click an Instance Name on the Manage IdP Adapter Instances screen.
3. Click Actions (if available).

To generate a properties list:
- Click Download under Action Invocation Link.

Extending an Adapter Contract

Adapters may be written with an option allowing administrators to add to the attributes that the adapter returns from a user's session. The PingFederate OpenToken Adapter, for example, provides such an option (see “OpenToken Adapter Configuration” on page 319).

To reach this screen:
1. Click Adapters on the Main Menu.
2. Click an Instance Name.
3. Click Extended Contract (if available).

To add an attribute:
- Enter the attribute name in the text box and click Add.
Chapter 5
Identity Provider SSO Configuration

Setting Pseudonym Values and Masking

On the Adapter Attributes screen you must select attributes to use for generating a pseudonym identifier (see “Account Linking” on page 5).

Optionally on this screen, you can also choose to mask the values of any or all attributes that PingFederate logs from this adapter instance at runtime (see “Attribute Masking” on page 9).

<table>
<thead>
<tr>
<th>Configuring IdP Adapter</th>
<th>Help</th>
<th>Support</th>
<th>About</th>
<th>Logout (Administrator)</th>
</tr>
</thead>
<tbody>
<tr>
<td>🔄 Main</td>
<td>Manage IdP Adapter Instances</td>
<td>Create Adapter Instance</td>
<td></td>
<td></td>
</tr>
<tr>
<td>✓ Type</td>
<td>✓ IdP Adapter</td>
<td>✓ Adapter Attributes</td>
<td>Summary</td>
<td></td>
</tr>
</tbody>
</table>

As an IdP, some of your SP partners may choose to receive a pseudonym to uniquely identify a user. From the attributes in this authentication adapter, please select the values that you would like to use in constructing this unique identifier. Optionally, specify here any attributes that must be masked in log files.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Pseudonym</th>
<th>Mask Log Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>subject</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Mask all OGNL-expression generated log values

To configure Pseudonym generation:
- Under Pseudonym select the value(s) to use.

Note: A selection is required regardless of whether you will use pseudonyms for account linking. This allows account linking to be used later without having to delete and reconfigure the adapter. Ensure that you choose at least one attribute that is unique for each user (for example, email) to prevent the same pseudonym from being assigned to multiple users.

To mask attributes in log files:
- Under Mask Log Values select the attribute(s) whose value(s) you want to mask.

If OGNL expressions might be used to map derived values into outgoing assertions and you want those values masked, select the related checkbox under the Attribute list (see “Using Attribute Mapping Expressions” on page 353).

To reach this screen:
1. Click Adapters on the Main Menu.
2. Click an Instance Name.
3. Click Adapter Attributes.

Selecting an Authentication Context

If you have deployed an integration kit that supports authentication context, you can specify the context in the IdP adapter configuration under Advanced Fields (see “Authentication Context” in the “Supported Standards” chapter of Getting Started).

For detailed information, see the OASIS SAML document: saml-authn-context-2.0-os.pdf.
To enter an authentication context URI for an adapter that supports this feature, click Advanced Fields on the adapter configuration screen.

**Editing and Saving Adapter Instances**

From the Adapter Instance Summary screen, you can reach adapter settings for editing.

**To edit the configuration:**
1. Click the heading above the information you want to change.
2. Make your changes.
3. Click Save on the configuration page and on the Manage IdP Adapter Instances screen.

**To save an adapter instance:**
1. Click Done on the Summary screen.
2. Click Save on the Manage IdP Adapter Instances screen.

**Configuring a Default URL and Error Message**

As an IdP, you can specify a default URL indicating a successful SLO to the end-user (if no other page is designated). On the IdP Default URL page, you can also customize an error message to be displayed as part of the error page rendered in the end-user's browser if an error occurs during IdP-initiated SSO. For example, you might consider modifying the default text to include useful information regarding whom the user should contact or what their next step should be.

**Note:** The error message is displayed only when the application calling the start-SSO endpoint does not explicitly provide its own error page URL.

Your application or your partner's application may supply the URL at runtime (see “IdP Endpoints” on page 335; but if none is provided, PingFederate will use the default value you enter on this screen.

**Tip:** If you leave the default URL blank, PingFederate provides a built-in landing page for the user. This Web page is among the templates you can modify with your own branding or other information (see “Customizing User-Facing Screens” on page 52).
Viewing Application Endpoints

Click Application Endpoints on the Main Menu to see a list of endpoints and descriptions applicable to your federation role (IdP or SP). These endpoints are built into PingFederate and cannot be changed.

Web-application developers at your site need to know the application endpoints to initiate transactions via PingFederate (see “SSO Integration Kits and Adapters” on page 4).

**Note:** For specific parameters required or allowed for Application Endpoints, see “IdP Endpoints” on page 335.

This screen also shows a Maintenance Endpoint that you can use to verify that the PingFederate server is running (see “Maintenance Endpoint” on page 343).

Viewing Protocol Endpoints

Click Protocol Endpoints under Federation Settings in the IdP Configuration section of the Main Menu to see a list of SAML, WS-Federation, and/or WS-Trust STS endpoints—a pop-up window displays only those endpoints related to the federation protocols you have chosen (see “Choosing Roles and Protocols” on page 64). These endpoints are built into PingFederate and cannot be changed.

Your federation partners or STS clients need to know the applicable IdP Services endpoints to communicate with your PingFederate server. Configured service endpoints for SAML connections are included in metadata export files (see “Exporting Metadata” on page 27).
The table below describes each endpoint:

<table>
<thead>
<tr>
<th>Service</th>
<th>URL and Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single Logout Service (SAML 2.0)</td>
<td>/idp/SLO.saml2</td>
</tr>
<tr>
<td></td>
<td>The URL that receives and processes logout requests and responses.</td>
</tr>
<tr>
<td>Single Sign-on Service (SAML 2.0)</td>
<td>/idp/SSO.saml2</td>
</tr>
<tr>
<td></td>
<td>The SAML 2.0 implementation URL that receives authentication requests for processing.</td>
</tr>
<tr>
<td>Artifact Resolution Service (SAML 2.0)</td>
<td>/idp/ARS.ssaml2</td>
</tr>
<tr>
<td></td>
<td>The SOAP endpoint that processes artifacts returned from a federation partner to retrieve the referenced XML message on the back channel.</td>
</tr>
<tr>
<td>Attribute Query Service (SAML 2.0)</td>
<td>/idp/attrsvc.ssaml2</td>
</tr>
<tr>
<td></td>
<td>The SAML implementation that receives and processes attribute requests.</td>
</tr>
<tr>
<td>Metadata Service</td>
<td>/</td>
</tr>
<tr>
<td></td>
<td>The default endpoint (empty path) from which partners can retrieve Auto-Connect metadata (see “Using Auto-Connect” on page 15).</td>
</tr>
<tr>
<td>Single Sign-on Service (SAML 1.x)</td>
<td>/idp/isx.saml1</td>
</tr>
<tr>
<td></td>
<td>The SAML 1.x implementation of IdP intersite transfer service (ISX) to which clients are redirected for SSO requests.</td>
</tr>
<tr>
<td>Artifact Resolution Service (SAML 1.x)</td>
<td>/idp/soap.ssaml1</td>
</tr>
<tr>
<td></td>
<td>The SOAP endpoint that processes artifacts returned from a federation partner to retrieve the referenced XML message on the back channel.</td>
</tr>
<tr>
<td>Single Sign-on Service (WS-Federation)</td>
<td>/idp/prp.wsf</td>
</tr>
<tr>
<td></td>
<td>The WS-Federation implementation URL that receives and processes security-token requests and SLO messages.</td>
</tr>
<tr>
<td>WS-Trust STS</td>
<td>/idp/sts.wst</td>
</tr>
<tr>
<td></td>
<td>The SOAP endpoint that receives and processes security-token requests from STS clients (Web Service Clients at the IdP site).</td>
</tr>
</tbody>
</table>
Managing SP Connections

As an IdP, you manage connection settings to support the exchange of federation-protocol messages (SAML, WS-Federation, or WS-Trust) with an SP or STS client application at your site.

**Note:** If you are configuring a new connection only for WS-Trust STS, follow the sections in this part of the manual up to and including “General Information” on page 117. Then turn to “WS-Trust STS Configuration” on page 269.

These settings include:

- User attributes you expect to send in an SSO assertion (including STS SAML tokens).
- User attributes that may be sent using the Attribute Query profile (if that profile is used).
- The protocol and, for SAML, the profile you will use, including detailed security specifications (the use of digital signatures, signature verification, XML encryption, and SSL). For more information see the “Supported Standards” chapter in Getting Started.

To continue with the configuration, you and your connection partner must have decided this information in advance (see “Federation Planning Checklist” on page 19). Your federation partner must supply some connection settings and other information (see “Configuration Data Exchange” on page 21).

**Tip:** PingFederate provides the capability of using a template to create simple, lightweight connections for browser-based SSO to SP partners (see “About PingFederate Express” on page 14).

**Tip:** If you are configuring connections to more than one partner under SAML 2.0 specifications, or if you intend to add partners in the future, consider using Auto-Connect (see “Configuring SP Auto-Connect” on page 189).

If your agreement includes sending assertions containing attribute values from a local data store, then you will need to define the data store during this configuration if you have not done so already (see “Managing Data Stores” on page 72).

Accessing Connections

You can create or modify connections directly via the Main Menu. Note that the menu displays the four most-recently modified connections. To view a list of all SP connections, click the Manage All SP link.

**Using the Main Menu**

From the Main Menu under My IdP Configuration, you can configure a new connection, modify an existing connection, or view connections.

**Tip:** To copy or delete connections or to recover connection drafts, click Manage All SP (see “Using the Manage Connections Screen” on page 111).
To begin configuring a new connection:

- Click **Create New** under **SP Connections** on the **Main Menu**.

**Tip:** If you want to use a virtual ID for a second connection to the same partner, the fastest way is to click **Manage All SP** and copy the first connection (see “Using the Manage Connections Screen” on page 111). For information about virtual IDs, see “Federation Server Identification” on page 20.

To modify a connection:

1. Click the connection name under **SP Connections** on the **Main Menu**.
   
   Only the four most recently edited connections are displayed. To see all connections, including drafts, click **Manage All SP**.

2. On the **Activation & Summary** screen, click the heading for the information you want to change.

3. Make your change and click **Save**.

**Note:** If **Save** is not available, it means your modification requires other changes or you are editing a screen that is part of a series of subtasks. Click **Next** and continue making indicated changes. The **Done** button indicates that further changes in the task are optional. When you have no further changes, click **Done** and then click **Save** on the task summary screen.

**Using the Manage Connections Screen**

From the Manage Connections screen you can configure a new connection, modify or copy an existing connection or draft, or delete a connection (if it is not active). In
addition, the screen provides export functions that allow you to save individual connection configurations or metadata about a connection that can be used to expedite your partner's corresponding configuration (see “Exporting Metadata” on page 27).

**Note:** The connection export function results in an XML file that you can modify and import into another PingFederate server acting in the same federation role (IdP or SP) at your site (see “Connection Management Service” on page 345). You can also automate this process (see “Automating Configuration Migration” on page 46).

Finally, from this screen you can also globally override transaction logging levels set for individual connections or restore connection-based logging (see “Runtime Transaction Logging” on page 26).

### Manage Connections

To access the Manage Connections screen:

- Click **Manage All SP** under **SP Connections** on the **Main Menu**.

To begin configuring a new SP connection:

- Click **Create Connection** on the Manage Connections screen.

**Tip:** If you need to create a second connection to a partner using a Virtual ID, copy the existing connection and make necessary changes, including adding the Virtual ID on the General Info screen. For information about Virtual IDs, see “Federation Server Identification” on page 20.

To copy a connection:

1. Click **Copy** under **Action** for the connection you want to copy.
2. Enter new General Information for the connection (see “General Information” on page 117).
3. Make any further changes needed for the new connection.

**Note:** SaaS-Provisioning configurations are not copied for a connection (see “SaaS Provisioning” on page 18).
To edit a connection or continue working on a draft:
► Click the Connection Name link.
   For a draft, you will return to where you left off.

To export a connection:
1. Click Export Connection under Action for the connection.
2. Save the XML file on your file system.
   You can change the name of the file, but keep the XML extension.

   **Tip:** You can import the connection programmatically or manually into another instance of PingFederate acting in the same role (see “Connection Management Service” on page 345).

To export connection metadata:
1. Click Export Metadata under Action for the connection.

   **Note:** For SP PingFederate Express connections, save the resulting ZIP file on your file system and send it your Express partner (see “About PingFederate Express” on page 14). For other connection types, continue to the next step.

   For non-Express connections this action takes you to the Export Metadata screen flow, with the connection selection preset (see “Exporting Metadata” on page 27).

   2. Complete the steps remaining in the Export Metadata screen flow (starting at Step 4 under “To export connection metadata:” on page 28).

To delete a connection:
1. Under Action, click Delete for the connection.
   (To undo the deletion, click Undelete.)

   **Note:** The Delete function is not available if the connection is active.

   2. To confirm the deletion, click Save.

To sort the list of connections:
► Click the arrow next to any column heading to sort the list based on that column.

To filter the list by Protocol and/or Status:
► Select a filter criterion from either or both of the drop-down lists.

To override connection-based transaction logging:
1. Select On under Logging Mode Override.
2. Choose the logging mode you want to use for all connections.

To restore connection-based transaction logging:
► Select Off under Logging Mode Override.
Choosing a Connection Template

On the Connection Template screen (shown only for a new connection), you can choose a quick-connection template for PingFederate Express partners (see “About PingFederate Express” on page 14). Or if your installation includes an optional PingFederate SaaS Connector, you can choose the template indicated for the applicable SaaS vendor (see “SaaS Provisioning” on page 18).

Tip: When you select a Connection Template, many connection settings are configured for you automatically. For information about using the PingFederate Express template and what steps remain to be configured, see “Using the Express Connection Template” on page 186). For information about using SaaS Connector templates, refer to the Quick Connection Guide available with your PingFederate SaaS Connector.

Choosing a Connection Type

If you are not using a connection template (which preconfigures browser-based SSO), indicate on the Connection Type screen whether the connection to this partner is for Browser SSO, WS-Trust STS, or both (see “Connection Types” on page 1).
If your federation deployment supports multiple protocols and you are not using a connection template, then for new SSO connections you can also select the applicable protocol on the **Connection Type** screen (see “Choosing Roles and Protocols” on page 64).

For secure Internet SSO, select **Browser SSO Profiles** and the **Protocol** (if necessary).

To configure a connection for WS-Trust STS, make that selection and then select a **Default Token Type**.

The **Default Token Type**, either SAML 1.1 or 2.0, is used when a Web Service client does not specify in the token request what token type the STS should issue.

Note: The **Default Token Type** does not need to match the **Protocol** indicated on the screen for SSO (when applicable).

(Optional) If your PingFederate license manages connections by groups, then you can select a group for this connection.

This option is not displayed for unrestricted or other types of licenses.

### Choosing Connection Options

On the **Connection Options** screen, if your installation includes a PingFederate SaaS Connector and you are using the associated connection template, you can confirm whether to configure provisioning (the default) for this connection (see “SaaS Provisioning” on page 18).
For SAML 2.0, you also have the option of configuring the Attribute Query profile, with or without SSO (see “Attribute Query and XASP” in the “Supported Standards” chapter of Getting Started).

Note: This screen is presented only for browser-based SSO connections (see “Choosing a Connection Type” on page 114).

To create a connection for secure Internet SSO, ensure that Browser SSO is selected and click Next.

If you are using a connection template for a SaaS provider but do not want to configure provisioning for this connection, clear the SaaS Provisioning checkbox.

### Importing Metadata

If you are using one of the SAML protocols (without a connection template) and have received a metadata file from your partner, click Browse on the Import Metadata screen, select the file, and click Next. For more information, see “Metadata” in the “Supported Standards” chapter of Getting Started.

Note: If the endpoints in the metadata file share the same base URL (protocol, hostname, and port), PingFederate uses this information to populate the Base URL field (see “General Information” on page 117). Consequently, individual endpoints on other screens do not include this information—only relative paths are shown.

Note: If you are importing a signed metadata file that does not include the certificate and public key, you will be asked to import the certificate needed to verify the XML signature (see the next section).

If you are not using a metadata file, click Next on the Import Metadata screen.

### Importing a Verification Certificate

The Import Certificate screen appears only if the metadata file you have chosen to import is signed and the certificate needed to verify the signature is not contained in the file.

Click Browse to locate and open the signature verification certificate for this partner.
Viewing the Metadata Summary

The Metadata Summary screen provides security information about an imported metadata file, including whether the file was signed and, if so, the trust status of the certificate used to verify the signature.

General Information

On the General Info screen, you provide a required unique identifier and display name for a connection, as well as optional contact information. In addition, on this screen you can set the level of transaction logging for this connection partner (see “Runtime Transaction Logging” on page 26).
Field Descriptions

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Partner’s Entity ID/ Audience/ Partner’s Realm (Connection ID)</td>
<td>(Required) The published, protocol-dependent, unique identifier of your partner. For a SAML 2.0 connection, this is your partner’s SAML Entity ID. For a SAML 1.x connection, this is the Audience your partner advertises. For a WS-Federation connection, this is your partner’s Realm. This ID may have been obtained out-of-band or via a metadata file if you are using a SAML protocol (see “Exporting Metadata” on page 27). For STS-only connections, this ID can be any unique identifier.</td>
</tr>
<tr>
<td>Connection Name</td>
<td>(Required) A plain-language identifier for the connection—for example, a company or department name. This name is displayed in the connection list on the Main Menu.</td>
</tr>
</tbody>
</table>
Managing SP Connections

To reach this screen:
1. Click a connection name on the Main Menu.
   Click Manage All SP, if needed, to see a full list of connections.
2. Click General Info under the SP Connection tab.

For a new connection:
- Fill in the information needed and click Next.
  - Connection ID and Connection Name are required (see “Field Descriptions” above).
  - Note that the Virtual ID identifies your own federation deployment for this connection only and overrides the ID you specified under Server Settings (see “Federation Server Identification” on page 20).

For an existing connection:
- If you are editing existing information, modify the fields as needed and click Save.

Configuring Browser-Based SSO

Browser-based SSO (also, Browser SSO) is another term for secure Internet SSO, which relies on a user’s Web browser and HTTP to broker XML identity-federation messaging between an IdP and an SP (in contrast to WS-Trust STS messaging, which is typically application-driven across the back channel and does not require browser mediation).

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Virtual Server ID</td>
<td>Enter a unique server ID in this field if you want to identify your server to this connection partner using an ID other than the one you specified under Server Settings (see “Specifying Federation Information” on page 67).</td>
</tr>
<tr>
<td>Base URL</td>
<td>The fully qualified hostname and port on which your partner’s federation deployment runs (e.g., <a href="https://www.pingidentity.com:9031">https://www.pingidentity.com:9031</a>). This entry is an optional convenience, allowing you to enter relative paths to specific endpoints, instead of full URLs, during the configuration process.</td>
</tr>
<tr>
<td>Company</td>
<td>The name of the partner company to which you are connecting.</td>
</tr>
<tr>
<td>Contact Name</td>
<td>The contact person at the partner company.</td>
</tr>
<tr>
<td>Contact Number</td>
<td>The phone number of the contact person at the partner company.</td>
</tr>
<tr>
<td>Contact Email</td>
<td>The email address for the contact person at the partner company.</td>
</tr>
<tr>
<td>Logging Mode</td>
<td>The level of transaction logging applicable for this connection (see “Runtime Transaction Logging” on page 26). Note that you can override connection logging mode settings globally from the connections list (see “Using the Manage Connections Screen” on page 111).</td>
</tr>
</tbody>
</table>
Chapter 5
Identity Provider SSO Configuration

To continue, click **Configure Browser SSO**.

**Configuration Steps**

Many steps involved in setting up a federation connection are protocol-independent; that is, they are required steps for all connections, regardless of the associated standards (see the “Supported Standards” chapter in Getting Started). Also, for any given connection, some configuration steps are required under the applicable protocol, while others are optional. Still others are required only based on certain selections. The PingFederate administrative console determines the required and optional steps based on the protocol and dynamically presents additional requirements or options based on selections.

The following sections provide sequential information about every step you might encounter while configuring browser-based SSO, regardless of the protocol you are using for a particular connection.

**Note:** The configuration screens represented in this chapter show “SAML 2.0” in their left corners, unless they are exclusive to WS-Federation or SAML 1.x setup requirements. When the SAML 2.0 screens are also applicable to SAML 1.x and/or WS-Federation connections, the SAML 2.0 representations and discussions also apply to the other protocols, unless otherwise indicated.

After configuring SSO settings, you will normally need to configure authentication credentials, the range of which depends on your SSO selections (see “Configuring Credentials” on page 160). Also, other configuration tasks may remain to be configured for new or modified connections, depending on selected connection options (see “Choosing Connection Options” on page 115).

**Important:** For new connections you must completely configure these SSO settings and subsequent tasks before you can save the connection on the Activation & Summary screen. Until then, the configuration is temporary and can be lost; the console times out after several minutes of inactivity. At any time, however, you can click **Save Draft**, which is available on most screens after you enter General Information (see “Console Buttons” in the “Console Navigation” chapter of Getting Started).

Use the lists and links (or page references) below to find specific information about steps that may apply to your SSO connection requirements:
SAML 2.0 SSO Configuration Steps

- “Choosing Profiles (SAML 2.0)” on page 121
- “Setting an Assertion Lifetime” on page 124
- “Assertion Creation” on page 124
  - “Choosing an Identity Mapping Method” on page 125
  - “Creating an Attribute Contract” on page 127
  - “IdP Adapter Mapping” on page 129
- “Configuring Protocol Settings” on page 147
  - “Setting Assertion Consumer Service URLs (SAML)” on page 148
  - “Specifying SLO Service URLs (SAML 2.0)” on page 150
  - “Choosing Allowable SAML Bindings (SAML 2.0)” on page 152
  - “Setting an Artifact Lifetime (SAML)” on page 152
  - “Specifying Artifact Resolver Locations (SAML 2.0)” on page 153
  - “Configuring Signature Policy” on page 154
  - “Configuring XML Encryption Policy (SAML 2.0)” on page 155

WS-Federation SSO Configuration Steps

- “Setting an Assertion Lifetime” on page 124
- “Assertion Creation” on page 124
  - “Choosing an Identity Mapping Method” on page 125
  - “Creating an Attribute Contract” on page 127
  - “IdP Adapter Mapping” on page 129
- “Configuring Protocol Settings” on page 147
  - “Defining a Service URL (WS-Federation)” on page 150

SAML 1.x SSO Configuration Steps

- “Setting an Assertion Lifetime” on page 124
- “Assertion Creation” on page 124
  - “Choosing an Identity Mapping Method” on page 125
  - “Creating an Attribute Contract” on page 127
  - “IdP Adapter Mapping” on page 129
- “Configuring Protocol Settings” on page 147
  - “Setting Assertion Consumer Service URLs (SAML)” on page 148
  - “Setting a Default Target URL (SAML 1.x)” on page 149
  - “Setting an Artifact Lifetime (SAML)” on page 152
  - “Configuring Signature Policy” on page 154

Choosing Profiles (SAML 2.0)

A SAML profile is the message-interchange scenario that you and your federation partner have agreed to use (see “Federation Planning Checklist” on page 19). For SAML 2.0, PingFederate supports all IdP- and SP-initiated SSO and SLO profiles.

For information on typical SSO/SLO profile configurations, including illustrations, see the “Profiles” sections in the “Supported Standards” chapter in Getting Started.
Identity Provider SSO Configuration

To reach this screen for editing:
1. Click a connection name on the Main Menu.
   - Click Manage All SP, if needed, to see a full list of connections.
2. Click Browser SSO under the SP Connection tab.
3. Click Configure Browser SSO.
4. Click SAML Profiles on the Summary screen.

To configure profiles:
1. Select the profile(s) applicable to this connection and click Next.
   - You must select an SSO profile before you can enable SLO.
2. Continue through the remaining connection-configuration tasks.

The following sections provide more information on requirements for each SAML profile:

Configuring IdP-Initiated SSO

When PingFederate is operating as an IdP, the IdP-initiated SSO profile configuration defines the message-transport mechanisms (bindings) your enterprise has agreed to use for this partner, plus optional digital signature requirements for outbound assertions (see “Certificates, SSL, and XML Encryption” on page 10).

For illustrations of typical profile and binding scenarios, see the “Profiles” sections in the “Supported Standards” chapter in Getting Started.

For this configuration you need to know:
- The transport binding(s) to which you and your partner have agreed

Note: This screen is not presented for SAML 1.x connections because IdP SSO is assumed, the SLO profiles are not supported, and the server supports SP-initiated SSO automatically (see “SAML 1.x Profiles” in the “Supported Standards” chapter of Getting Started).

The screen is also not presented for WS-Federation connections because profile selection is not required (see “WS-Federation” in the “Supported Standards” chapter of Getting Started).
Managing SP Connections

- The certificate to be used for signing assertions (not always required for the artifact binding)
- The URL(s) of your partner's Assertion Consumer Service(s)

**Configuring SP-Initiated SSO**

The SP-initiated profile configuration for SSO defines the message-transport mechanisms (bindings) and security requirements for receiving authentication requests and sending assertions when your SP partner initiates SSO transactions (see “Single Sign-on” in the “Supported Standards” chapter of *Getting Started*).

For SAML 1.x, the SP-initiated SSO profile is also known as the “destination-first” profile, which was added as a supported “non-normative” use case after the release of the SAML 2.0 specifications. As an IdP, you do not need to configure this profile; when the SP sends an authentication request to your SSO Service endpoint, PingFederate will handle the response automatically.

For illustrations of typical profile and binding scenarios, see the “Profiles” sections in the “Supported Standards” chapter in *Getting Started*.

For this configuration you will need to know:
- The endpoint URL(s) for your SP’s Assertion Consumer Service(s)
- The transport bindings that you and your partner have agreed upon inbound and outbound
- The certificates you will use to sign outbound assertions and to verify incoming digital signatures from your SP, when either is required

When Artifact is an allowable inbound SAML binding, you also need to know the endpoint(s) to your partner's Artifact Resolution Service(s) and the SOAP client authentication mechanism to use: either HTTP Basic, SSL client certificates, a digital signature, or a combination of these mechanisms.

**Configuring IdP-Initiated SLO**

The SAML 2.0 IdP-initiated SLO profile configuration defines the message-transport mechanisms (bindings) and security requirements for exchanging SLO requests and responses.

For more information about SLO, see “Single Logout” in the “Supported Standards” chapter of *Getting Started*.

For this configuration you need to know:
- The transport bindings to which you and your partner have agreed to send SLO requests and receive responses
- The certificates to be used for signing outgoing messages and for verifying incoming digital signatures from your SP (not always required for the artifact binding)
- The URL(s) of your SP’s Single Logout Service(s)

**Configuring SP-Initiated SLO**

The SAML 2.0 SP-initiated SLO profile configuration defines the message-transport mechanisms (bindings) and security requirements that you and your partner have agreed upon for exchanging SAML requests and responses.
Chapter 5
Identity Provider SSO Configuration

For more information about SLO, see “Single Logout” in the “Supported Standards” chapter of Getting Started.

For this configuration you need to know:

- The transport bindings that you and your partner have agreed upon to send SLO requests and receive responses
- The certificates to be used for signing outgoing messages and for verifying incoming digital signatures from your SP (not always required for the artifact binding)
- The URL(s) of your SP’s Single Logout Service(s)
- The URL of your SP’s Artifact Resolution Service(s)—if this binding and endpoint are different from your SSO configuration—and SOAP client authentication requirements

Setting an Assertion Lifetime

Identity-federation standards require a window of time during which an assertion is considered valid. Each assertion has a time-stamp XML element as well as elements indicating the allowable lifetime of the assertion (in minutes) before and after the assertion time stamp.

Field Descriptions

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minutes Before</td>
<td>The amount of time before the assertion was issued during which it is to be considered valid.</td>
</tr>
<tr>
<td>Minutes After</td>
<td>The amount of time after the assertion was issued during which it is to be considered valid.</td>
</tr>
</tbody>
</table>

To change the default times:

- (Optional) Edit the desired setting(s) and click **Next** or **Save**.

Assertion Creation

As an IdP, you must specify how PingFederate will obtain user-authentication information and use it to create assertions appropriate for your SP partner, including additional user attributes as needed. This configuration includes:

- Choosing an identity-mapping method (see “Choosing an Identity Mapping Method” next).
• Defining the attribute contract you will use with this partner, if any (see "Creating an Attribute Contract" on page 127).

• Configuring instances of one or more adapters (see "IdP Adapter Mapping" on page 129) and specifying how they are used to fulfill the attribute contract.

<table>
<thead>
<tr>
<th>SAML SP Connection</th>
<th>Help</th>
<th>Support</th>
<th>About</th>
<th>Logout (Administrator)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SP Connection</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Browser SSO</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Click Configure Assertion Creation to continue.

Choosing an Identity Mapping Method

PingFederate allows your SP partner to use either account linking or account mapping to associate remote users with local accounts for SSO between business partners (see "Identity Mapping" on page 5). At the Identity Mapping step, you choose the type of name identifier your partner needs to facilitate one of these options. You and your partner may want to decide in advance which option to use (see "Federation Planning Checklist" on page 19).

The choices of name-identifier types depend on which protocol you are using:

• For information about SAML selections, see “SAML Name ID Selections” next.

• For information about WS-Federation selections, see “WS-Federation Name ID Selections” on page 127.

SAML Name ID Selections

The allowable types of name identifiers for SAML connections are described below. These choices affect how SPs make use of account mapping or account linking.
If your SP is using account linking, then establishing an attribute contract is not required. Depending on your partner agreement, however, you may choose to supplement the account link with an attribute contract. In this configuration the account link is used to determine the user's identity, while the additional attributes might be used for authorization decisions, customized Web pages, and so on, at the SP site (see “About Attributes” on page 6).

Important: If you have previously set up a configuration to use an attribute contract and want to change the configuration to use account linking without additional attributes, then the existing attribute contract will be discarded.

Account linking can be used with either a clear, standard name identifier or an opaque pseudonym.

- If you want to send a known attribute to identify a user—for example, a username or email address—then select **Standard**.

- If you and your partner have agreed to use an opaque persistent name identifier (often used for account linking), then select **Pseudonym** on the Identity Mapping screen.

  The pseudonym is based on values pulled from the IdP adapter instance used to authenticate the user. You select these values when you configure IdP adapter instances (see “Setting Pseudonym Values and Masking” on page 106).

  To set up an attribute contract to use in conjunction with an opaque identifier, click the checkbox next to “Include attributes . . .” after selecting **Pseudonym**.

- Select **Transient** to enhance the privacy of a user's identity. Unlike a pseudonym, a transient identifier is different each time a user initiates SSO (see “Account Linking” on page 5).

  A typical application for this selection might be, for example, when an SP provides generalized group accounts based on organizational rather than individual identity.

  To set up an attribute contract to use in conjunction with an opaque identifier, click the checkbox next to “Include attributes . . .” after selecting **Transient**.
Managing SP Connections

To reach this screen for editing:
1. Click the connection name on the Main Menu.
   Click Manage All SP, if needed, to see a full list of connections.
2. Click Browser SSO under the SP Connection tab.
3. Click Configure Browser SSO.
4. Click Assertion Creation under the Browser SSO tab.
5. Click Configure Assertion Creation.
6. Click Identity Mapping on the Summary screen.

WS-Federation Name ID Selections
For WS-Federation purposes a name identifier is a uniquely identifying user attribute.

- **Email Address**: This attribute is commonly used as a unique identifier for SSO and SLO. Make this selection, for example, if a user logs in using an email address or if the information is available for lookup in a local data store.

- **User Principal Name**: The username or other unique ID of the subject initiating the transaction. Make this selection, for example, if a username will be available from the current user session as part of a cookie or can be derived from a local data store.

- **Common Name**: This selection provides for anonymous SSO to your SP, generally using a hard-coded generalized logon. Make this selection if your partner agreement involves a many-to-one use case—for example, if the SP has a group account set up for all users in a particular domain.

Later, you will map your choice to the SAML_SUBJECT attribute in the SAML assertion (see “Mapping Default Attribute Contract Fulfillment” on page 145).

Creating an Attribute Contract
An attribute contract is the set of user attributes that you and your partner have agreed will be sent in SAML assertions for this connection (see “Attribute Contracts” on page 7). You identify these attributes on this screen.

If you are sending a “standard” name identifier (see “Choosing an Identity Mapping Method” on page 125), then the contract includes the default SAML_SUBJECT, which identifies the user in the assertion. You will configure this variable later to contain a user ID or another agreed-upon attribute—for instance, an email address—that uniquely identifies the user (see “Attribute Contract Fulfillment” on page 142).

**Note**: Creating an attribute contract is optional if you are sending either a pseudonym or a transient identifier to your connection partner (see “Choosing an Identity Mapping Method” on page 125).
To reach this screen for editing:

1. Click the connection name on the Main Menu. Click Manage All SP, if needed, to see a full list of connections.
2. Click Browser SSO under the SP Connection tab.
3. Click Configure Browser SSO.
4. Click Assertion Creation under the Browser SSO tab.
5. Click Configure Assertion Creation.
6. Click Attribute Contract on the Summary screen.

If this step is not in the list, then you have chosen to send either a pseudonym or a transient identifier without additional attributes (see “Choosing an Identity Mapping Method” on page 125).

**Note:** This screen is different for SAML 1.x and WS-Federation configurations. Attribute Name Format is not part of the specifications. In addition, for WS-Federation, the Subject Name Format is determined by the previous screen (see “WS-Federation Name ID Selections” on page 127).

7. **To add an attribute:**

1. Enter the attribute name in the text box.

   Attribute names are case-sensitive and must correspond to the names expected by your partner.

   **Tip:** You can add a special attribute, SAML\_AUTHN\_CTX, to indicate to the SP (if required) the type of credentials used to authenticate to the IdP application (see “Authentication Context” in the “Supported Standards” chapter of Getting Started). Map a value for the authentication context on the attribute-mapping screen later in the configuration, from any available attribute source (see “Attribute Contract Fulfillment” on page 142).

2. (Optional) For SAML 2.0 connections, select the Attribute Name Format.

   Normally, the default selection is applicable, unless you and your partner have specifically agreed to other format specifications (see “Name Formats” on page 7).

3. Click Add.
(Optional) For SAML connections, to modify the Subject Name Format:

- Make your selection in the drop-down list and click Done.

  The default selection is usually applicable, unless you and your partner have agreed to a different format specification (see “Name Formats” on page 7).

**Note:** This selection is not available if you are sending a pseudonym as the subject, or a transient identifier (see “Choosing an Identity Mapping Method” on page 125).

To modify an attribute name or format:

1. Click Edit under Action for the attribute.
2. Make the change and click Update.

**Note:** If you change your mind, ensure that you click the Cancel link in the Actions column, not the Cancel button, which discards any other changes you might have made in the configuration steps.

To delete an attribute:

- Click Delete under Action for the attribute.

**IdP Adapter Mapping**

IdP adapters are responsible for handling user authentication as part of an SSO operation (see “SSO Integration Kits and Adapters” on page 4). A configured and deployed adapter in PingFederate is known as a adapter instance. The same instance may be mapped by multiple connections.

Map one or more IdP adapter instances into each SP connection so that when a user authenticates with a particular external identity management system the user attributes are returned to PingFederate.

Regardless of how many IdP adapter instances are mapped in an SP connection, PingFederate uses only one instance to authenticate a user. Because each instance may return different user attributes, each IdP adapter mapping must define how the attribute contract is fulfilled; you must map attributes from each adapter—and/or attributes retrieved from your local data stores—into the assertions PingFederate will send to this SP to fulfill the attribute contract.

You begin this configuration on the IdP Adapter Mapping screen, where you choose to map instances of IdP adapters. If you have not yet configured an instance of the adapter you intend to use within this SP connection, see “Configuring IdP Adapters” on page 102.
To reach this screen for editing:
1. Click the connection name on the Main Menu.
   - Click Manage All SP, if needed, to see a full list of connections.
2. Click Browser SSO under the SP Connection tab.
3. Click Configure Browser SSO.
4. Click Assertion Creation under the Browser SSO tab.
5. Click Configure Assertion Creation.
6. Click IdP Adapter Mapping on the Summary screen.

To modify an existing Adapter Instance:
1. Click its Name link.

To begin configuring an Adapter Instance for this connection:
- Click Map New Adapter Instance.

Selecting an Adapter Instance

A configured and deployed adapter in PingFederate is known as an adapter instance. The same adapter instance may be mapped by multiple connections (see “Configuring IdP Adapters” on page 102).

You can use attributes returned from the adapter (the adapter contract) to fulfill the attribute contract with this partner, and/or use them to look up additional attributes in a user data store. You make this choice on the next screen (see “Selecting Assertion Mapping” on page 131).
Choose an Adapter Instance from the drop-down list and click Next to continue.

To create or change an adapter instance, as needed, click Manage Adapter Instances.

To reach this screen for editing:
1. Click the connection name on the Main Menu.
   Click Manage All SP, if needed, to see a full list of connections.
2. Click Browser SSO under the SP Connection tab.
3. Click Configure Browser SSO.
4. Click Assertion Creation under the Browser SSO tab.
5. Click Configure Assertion Creation.
6. Click IdP Adapter Mapping on the Summary screen.

If this step is not available, you have chosen to pass either a pseudonym or transient name identifier and not to use additional attributes (see “Choosing an Identity Mapping Method” on page 125).

Selecting Assertion Mapping

For SAML assertions, you can query local user data stores to help fulfill the attribute contract, in conjunction with attribute values supplied by the IdP adapter you are using with PingFederate (see “SSO Integration Kits and Adapters” on page 4).

The values supplied by the adapter you have chosen are shown under Adapter Contract on the Assertion Mapping screen.
Chapter 5
Identity Provider SSO Configuration

To reach this screen for editing:

1. Click the connection name on the Main Menu.
   - Click Manage All SP, if needed, to see a full list of connections.
2. Click Browser SSO under the SP Connection tab.
3. Click Configure Browser SSO.
4. Click Assertion Creation under the Browser SSO tab.
5. Click Configure Assertion Creation.
6. Click IdP Adapter Mapping on the Summary screen.
   - If this step is not available, you have chosen to pass either a pseudonym or transient name identifier and not to use additional attributes (see “Choosing an Identity Mapping Method” on page 125).
7. Click the Adapter Instance Name.
8. Click Assertion Mapping on the Summary screen.
   - If you choose to retrieve additional attributes, then you will identify data stores and specify lookup queries next (see “Configuring Attribute Sources and User Lookup” on page 133).
   - If you use only the Adapter Contract values, then you will map values for the attribute contract next (see “Mapping Default Attribute Contract Fulfillment” on page 145).

Tip: To determine whether you need to look up additional values, compare the attribute contract against the adapter contract (see “Creating an Attribute Contract” on page 127 and “IdP Adapter Mapping” on page 129). If the attribute contract requires more information, determine whether local data stores can supply it. (You can also choose to use text constants or expressions for certain information—see “Mapping Default Attribute Contract Fulfillment” on page 145.)
Configuring Attribute Sources and User Lookup

Attribute sources are specific database or directory locations containing information that may be needed for the attribute contract (see “Creating an Attribute Contract” on page 127). Attribute sources can be reused across connections to other SP partners.

This portion of the connection configuration allows you to set up search parameters for your data stores, including “fall-through” searches. For example, you can add the same data store more than once, using different search queries for each instance, or you can search different data stores successively, either with the same queries (for failover) or different ones.

If any search fails to find a user in the specified Attribute Source, the next search is executed until a match is found.

**Note:** Queries are executed in the order of Attribute Sources shown. Use the move up/move down controls as needed to adjust the order. Note, however, that data can originate from only one source.

To reach this screen for editing:

1. Click the connection name on the Main Menu.
   
   Click Manage All SP, if needed, to see a full list of connections.

2. Click Browser SSO under the SP Connection tab.

3. Click Configure Browser SSO.

4. Click Assertion Creation under the Browser SSO tab.

5. Click Configure Assertion Creation.

6. Click IdP Adapter Mapping on the Summary screen.
   
   If this step is not available, you have chosen to pass either a pseudonym or transient name identifier and not to use additional attributes (see “Choosing an Identity Mapping Method” on page 125).

7. Click the Adapter Instance Name.

8. Click Attribute Sources & User Lookup under the IdP Adapter Mapping tab.
   
   If this step is not listed, then this instance is configured to use adapter values only (see “Selecting Assertion Mapping” on page 131).

To configure an attribute source:

- Click Add Attribute Source and complete the setup steps (see “Attribute Source Setup” next).
To modify an attribute source configuration:

1. Click the attribute source Description link.
2. Click **Save** on the screen you change.

**Note:** Depending on what you change, you may need to modify dependent data in subsequent steps, as indicated. Click **Save** or **Done** when either of those options appears.

**Attribute Source Setup**

For attribute-source setup information, refer to the sections indicated in the following steps.

**Note:** As you make selections on configuration screens, ensure that you allow enough time for PingFederate to access your data store and populate drop-down lists.

1. See “Selecting a Data Store” (next section).
2. See the following sections in this manual, depending on the type of data store:

<table>
<thead>
<tr>
<th>Data Store Type</th>
<th>Related Manual Sections</th>
</tr>
</thead>
</table>
| JDBC            | • “Selecting a JDBC Database Table and Columns” on page 135  
|                 | • “Configuring a Database Filter (WHERE Clause)” on page 137 |
| LDAP            | • “Configuring an LDAP Directory Search” on page 139  
|                 | • “Configuring an LDAP Filter” on page 140 |
| Custom          | • “Configuring Custom Source Filters” on page 142  
|                 | • “Selecting Custom Source Fields” on page 142 |

3. See “Attribute Contract Fulfillment” on page 142.

**Selecting a Data Store**

This screen allows you to choose a data store from a previously configured list (see “Managing Data Stores” on page 72). Attribute values extracted from this data store will be used to help fulfill the attribute contract for this partner (see “Creating an Attribute Contract” on page 127).
To reach this screen for editing:

1. Click the connection name on the Main Menu.
   - Click Manage All SP, if needed, to see a full list of connections.
2. Click Browser SSO under the SP Connection tab.
3. Click Configure Browser SSO.
4. Click Assertion Creation under the Browser SSO tab.
5. Click Configure Assertion Creation.
6. Click IdP Adapter Mapping on the Summary screen.
   - If this step is not available, you have chosen to pass either a pseudonym or transient name identifier and not to use additional attributes (see “Choosing an Identity Mapping Method” on page 125).
7. Click the Adapter Instance Name.
8. Click Attribute Sources & User Lookup under the IdP Adapter Mapping tab.
   - If this step is not shown, you have elected not to look up attributes in data stores (see “Selecting Assertion Mapping” on page 131).
9. Click the attribute source Description link.

To choose a Data Store:

- Choose an Active Data Store and click Next.

A data store configuration must be defined under System Settings for use within a connection. If the data store you want is not shown in the drop-down menu, click Manage Data Stores to add a new data store (see “Managing Data Stores” on page 72).

Selecting a JDBC Database Table and Columns

When you choose to use a database source for attributes, you follow this path through the configuration steps.

On this screen you begin to specify exactly where additional data can be found to complete the attribute contract when you send an assertion to this SP (see “Creating an Attribute Contract” on page 127). Only one table may be used as a source of data for a JDBC lookup.
Field Descriptions

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schema</td>
<td>Lists the table structure that stores information within a database. Some databases, such as Oracle, require selection of a specific schema for a JDBC query. Other databases, such as MySQL, do not require selection of a schema.</td>
</tr>
<tr>
<td>Table</td>
<td>The name of the table contained in the database. Use the drop-down to change the table.</td>
</tr>
<tr>
<td>Columns to return from SELECT</td>
<td>Displays selected table columns. Select the columns that are associated with the desired attributes you would like to return from the JDBC query.</td>
</tr>
</tbody>
</table>

To reach this screen for editing:

1. Click the connection name on the Main Menu.
   - Click Manage All SP, if needed, to see a full list of connections.
2. Click **Browser SSO** under the SP Connection tab.
3. Click **Configure Browser SSO**.
4. Click **Assertion Creation** under the Browser SSO tab.
5. Click **Configure Assertion Creation**.
6. Click **IdP Adapter Mapping** on the Summary screen.
   - If this step is not available, you have chosen to pass either a pseudonym or transient name identifier and not to use additional attributes (see “Choosing an Identity Mapping Method” on page 125).
7. Click the Adapter Instance Name.
8. Click **Attribute Sources & User Lookup** under the IdP Adapter Mapping tab. If this step is not shown, you have elected not to look up attributes in data stores (see “Selecting Assertion Mapping” on page 131).

9. Click the attribute source Description link.

10. Click **Database Table and Columns**.

**To select a database table and columns for queries:**

1. Choose a **Schema** file (when applicable) from the drop-down list.
2. Choose a **Table** from the drop-down list.
3. Choose a name under **Columns to Return from Select** and click **Add Column**. Repeat this step for other columns as needed.

---

**Note:** You do not need to add a column here for it to be used as part of a search key (see “Configuring a Database Filter (WHERE Clause)” next). Add only attributes from which you need actual values to pass in an assertion.

---

**Tip:** To determine what attributes to look up during a query, click the **View Attribute Contract** link to see what information must be collected (see “Creating an Attribute Contract” on page 127). Then determine what information is coming in from the session lookup adapter (see “Selecting Assertion Mapping” on page 131). Information not contained in the adapter contract may be pulled from the data store look-up query.

**Configuring a Database Filter (WHERE Clause)**

The JDBC **WHERE** clause in PingFederate queries the data table you selected to retrieve a record associated with a particular value (or values) from the assertion. The clause is in the form:

```
WHERE column1=value1 [AND column2=value2] [0...]
```

The left side of the first variable pair uses a column name in the database table you selected (see “Selecting a JDBC Database Table and Columns” on page 135).

The right side generally uses values passed in from your session lookup adapter (variables, including the correct formatting, are listed under Adapter Values—see “Configuring IdP Adapters” on page 102).

You can also apply additional search criteria from your own database, using any other columns from the targeted table.

**Tip:** Click “**View List of Columns . . .**” to see a list from which to copy and paste.

For general information about **WHERE** clauses, consult your DBMS documentation.
EXAMPLE:

```
userid='${username}'
```

In this example, `userid` is the name of a column in the JDBC data store. On the right side, `'${username}'` returns the value of the `username` variable from the IdP adapter.

**Important:** You must use the `$()` syntax to retrieve the value of the enclosed variable and use single quotation marks around the `$()` characters.

---

**Field Descriptions**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Where</td>
<td>WHERE clause statements conditionally select data from a table. Enter the WHERE clause statement in the space provided. For example: <code>WHERE email='clive@company.com'</code>.</td>
</tr>
</tbody>
</table>

**To reach this screen for editing:**

1. Click the connection name on the Main Menu.
2. Click **Manage All SP**, if needed, to see a full list of connections.
3. Click **Browser SSO** under the SP Connection tab.
4. Click **Configure Browser SSO**.
5. Click **Assertion Creation** under the Browser SSO tab.
6. Click **Configure Assertion Creation**.
6. Click **Attribute Sources & User Lookup** under the IdP Adapter Mapping tab. If this step is not shown, you have elected not to look up attributes in data stores (see “Selecting Assertion Mapping” on page 131).

7. Click the attribute source Description link.

8. Click **Database Filter** from the steps list.

**To construct the WHERE clause:**

1. Enter the statement in the space provided, following the guidelines and example above. The initial WHERE is optional.

2. Ensure the syntax and variable names are correct.

   When you click **Next**, you will map attribute values returned from the database into the assertion (see “Attribute Contract Fulfillment” on page 142).

### Configuring an LDAP Directory Search

When you choose to use an LDAP source for attributes, you follow this path through the configuration steps.

On this screen you specify the branch of your LDAP hierarchy where you want PingFederate to look up user data.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base DN</td>
<td>The base distinguished name of the tree structure in which the search begins. This field is optional if records are located at the LDAP root.</td>
</tr>
<tr>
<td>Search Scope</td>
<td>Determines the node depth of the query. Select Subtree, One level or Object.</td>
</tr>
<tr>
<td>Root Object Class</td>
<td>The class containing the attributes you want.</td>
</tr>
</tbody>
</table>
Chapter 5
Identity Provider SSO Configuration

To reach this screen for editing:
1. Click the connection name on the Main Menu.
   Click Manage All SP, if needed, to see a full list of connections.
2. Click Browser SSO under the SP Connection tab.
3. Click Configure Browser SSO.
4. Click Assertion Creation under the Browser SSO tab.
5. Click Configure Assertion Creation.
6. Click IdP Adapter Mapping on the Summary screen.
   If this step is not available, you have chosen to pass either a pseudonym or
transient name identifier and not to use additional attributes (see “Choosing an
Identity Mapping Method” on page 125).
7. Click the Adapter Instance Name.
8. Click Attribute Sources & User Lookup under the IdP Adapter Mapping tab.
   If this step is not shown, you have elected not to look up attributes in data stores
   (see “Selecting Assertion Mapping” on page 131).
9. Click the attribute source Description link.
10. Click LDAP Directory Search from the steps list or fill out the appropriate
    screens and advance to this screen.
    If you have not yet defined an LDAP data store, see “Selecting a Data Store” on
    page 134.

To select LDAP attributes:
1. (Optional) Enter a Base DN.
2. Select a Search Scope.
3. Select a Root Object Class.
4. Under Attributes to return from search, choose an attribute and click Add
   Attribute.
   Note that the attribute Subject DN is always returned by default.
5. Repeat the last step for other attributes as needed.
6. (Optional) Change the Search Scope or the Root Object Class if you want
   attributes from other locations.

   Note: You do not need to add an attribute here for it to be used in a
   search filter (see “Configuring an LDAP Filter”). Add only attributes
   from which you need actual values to pass in an assertion.

Configuring an LDAP Filter

The LDAP filter queries the data you selected to retrieve a record associated with a
particular value (or values) from the user's session. The filter is in the form:

\( (attribute=${value}) \)
Managing SP Connections

The left-side variable is an attribute you selected earlier (see “Configuring an LDAP Directory Search” on page 139).

The right side generally uses values passed in from your session lookup adapter (variables, including the correct syntax, are listed under Adapter Values—see “Configuring IdP Adapters” on page 102).

You can also apply additional search criteria from your data store, using any other attributes from the targeted object classes.

**Tip:** Click “View List of Available LDAP Attributes” for a list from which you can copy and paste.

For general information about search filters, consult your LDAP documentation.

---

### Field Descriptions

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filter</td>
<td>Narrows a search to locate requested data by either including or excluding specific records. An LDAP filter includes the attributes in the search and the value or range of values that the search is attempting to match. Searches are conducted by using three components: 1) at least one attribute (attribute data type) to search on, 2) a search filter operator that will determine what to match, and 3) the value of the attribute being sought. Searches must have at least one of each of these three components.</td>
</tr>
</tbody>
</table>

To reach this screen for editing:

1. Click the connection name on the Main Menu.
   
   Click Manage All SP, if needed, to see a full list of connections.

2. Click Browser SSO under the SP Connection tab.
3. Click Configure Browser SSO.
4. Click Assertion Creation under the Browser SSO tab.
5. Click Configure Assertion Creation.
Chapter 5
Identity Provider SSO Configuration

6. Click **IdP Adapter Mapping** on the Summary screen.
   If this step is not available, you have chosen to pass either a pseudonym or transient name identifier and not to use additional attributes (see “Choosing an Identity Mapping Method” on page 125).

7. Click the Adapter Instance Name.

8. Click **Attribute Sources & User Lookup** under the IdP Adapter Mapping tab.
   If this step is not shown, you have elected not to look up attributes in data stores (see “Selecting Assertion Mapping” on page 131).

9. Click the attribute source Description link.

10. Click **LDAP Filter** from the steps list.
    If you have not yet defined an LDAP data store, see “Selecting a Data Store” on page 134.

**To construct the LDAP filter:**

1. Enter the statement in the space provided, following the guidelines and example above.

   **Note:** If you used an anonymous binding to create this LDAP connection, your access might be restricted (see “Configuring an LDAP Connection” on page 77).

2. Ensure the syntax and variable names are correct.

3. Click **Next**.

**Configuring Custom Source Filters**

When you choose to use a custom source for attributes, you follow this path through the configuration steps.

On this screen you specify a filter, or lookup query, for your custom data source. This screen display and the syntax of the filter depends on your developer's implementation of the custom source SDK.

**Selecting Custom Source Fields**

On the Configure Custom Source Fields screen, you can choose from among the fields shown to map to the adapter contract. These choices are supplied by the driver implementation. Select only those needed to fulfill the attribute contract for this partner connection.

**Attribute Contract Fulfillment**

The last step in configuring an attribute source is to map values into the attribute contract (see “Creating an Attribute Contract” on page 127). These are the values that will be included in assertions sent to this SP (provided the information is found in this attribute source).

You map attributes on the Attribute Contract Fulfillment screen.
Map each attribute to fulfill the Attribute Contract from one of these Sources:

- **Adapter**
  Values are returned from the session. When you make this selection, the associated Value drop-down list is populated by the session-lookup adapter (see “SSO Integration Kits and Adapters” on page 4).

  For example, you might choose the adapter attribute `username` to map to `SAML_SUBJECT`.

- **LDAP/JDBC/Custom**
  Values are returned from your attribute source. When you make this selection, the Value list is populated by the LDAP, JDBC or Custom attributes you identified for this Attribute Source (see “Configuring an LDAP Directory Search” on page 139, “Selecting a JDBC Database Table and Columns” on page 135, or “Configuring Custom Source Filters” on page 142).

- **Expression (when enabled)**
  This option provides more complex mapping capabilities—for example, transforming incoming values into different formats (see “Using Attribute Mapping Expressions” on page 353). All of the variables available for text entries (see below) are also available for expressions.

- **Text**
  The value is what you enter. This can be text only, or you can mix text with references to any of the values from the adapter, using the `${attribute}` syntax.

  You can also enter values from your data store, when applicable, using this syntax:
  
  `${ds.attribute}`

  where `attribute` is any of the data store attributes you have selected.

  **Tip:** Two other variables are also available: `${SAML_SUBJECT}` and `${TargetResource}`. `${SAML_SUBJECT}` is the initiating user (or other entity). `${TargetResource}` is a reference to the protected application or other resource for which the user requested SSO access; this variable is available only if specified as a query parameter for the relevant PingFederate endpoint (either as `TargetResource` for SAML 2.0 or `TARGET` for SAML 1.x—see “Application Endpoints” on page 335).
There are a variety of reasons why you might hard code a text value. For example, if your SP's Web application provides a service based on your company's name, you might provide that attribute value as a constant.

**To reach this screen for editing:**
1. Click the connection name on the Main Menu.
   - Click Manage All SP, if needed, to see a full list of connections.
2. Click Browser SSO under the SP Connection tab.
3. Click Configure Browser SSO.
4. Click Assertion Creation under the Browser SSO tab.
5. Click Configure Assertion Creation.
6. Click IdP Adapter Mapping on the Summary screen.
   - If this step is not available, you have chosen to pass either a pseudonym or transient name identifier and not to use additional attributes (see “Choosing an Identity Mapping Method” on page 125).
7. Click the Adapter Instance Name.
8. Click Attribute Sources & User Lookup under the IdP Adapter Mapping tab.
   - If this step is not shown, you have elected not to look up attributes in data stores (see “Selecting Assertion Mapping” on page 131).
9. Click the attribute source Description link.
10. Click Attribute Contract Fulfillment from the steps list.
    - If you have not yet defined a data store, see “Selecting a Data Store” on page 134).

**To map attributes:**
1. Choose a Source for each Target attribute.
   - See “Map each attribute to fulfill the Attribute Contract from one of these Sources:” on page 143.
2. Choose (or enter) a Value for each Attribute.
   - All values must be mapped.
3. Click Next.

**Using the Attribute Source Summary Screen**
When you have finished configuring Attribute Sources and User Lookup, you can review the configuration on the Summary screen.

If you need to make any changes, click the heading over the information you want to edit. When you are finished, click Done to continue with IdP Adapter Mapping configuration. If you are editing an existing connection, click Done on successive screens until you reach the Assertion Creation screen and then click Save.

**Specifying a Failsafe Attribute Source**
If attributes needed to fulfill an attribute contract cannot be found in your data stores, you can either map a default set of attribute values to send—identifying a “guest” user, for example—or you can have PingFederate stop the SSO transaction. This choice depends on your agreement with the SP.
To reach this screen for editing:

1. Click the connection name on the Main Menu.
   - Click Manage All SP, if needed, to see a full list of connections.

2. Click Browser SSO under the SP Connection tab.

3. Click Configure Browser SSO.

4. Click Assertion Creation under the Browser SSO tab.

5. Click Configure Assertion Creation.

6. Click IdP Adapter Mapping on the Summary screen.
   - If this step is not available, you have chosen to pass either a pseudonym or transient name identifier and not to use additional attributes (see “Choosing an Identity Mapping Method” on page 125).

7. Click the Adapter Instance Name.

8. Click Failsafe Attribute Source on the Summary screen.
   - This step appears only if you are using data stores (see “Selecting Assertion Mapping” on page 131).

To specify whether to use a failsafe attribute source:

- Make the relevant selection to use either a default set of attributes or to terminate the SSO, and then click Next.

Mapping Default Attribute Contract Fulfillment

Fulfillment of the attribute contract must be specified whether or not data sources are used. You accomplish this on the Attribute Contract Fulfillment screen, either by choosing to configure default mappings after setting up attribute sources (see “Specifying a Failsafe Attribute Source” on page 144) or if you choose not to set up attribute sources (see “Selecting Assertion Mapping” on page 131).

Map each attribute to fulfill the Attribute Contract from one of these Sources:

- Adapter
  - Values are returned from the session. When you make this selection, the associated Value drop-down list is populated by the session-lookup adapter (see “SSO Integration Kits and Adapters” on page 4).
  - For example, you might choose the adapter attribute username to map to SAML_SUBJECT.
• Expression (when enabled)

This option provides more complex mapping capabilities—for example, transforming incoming values into different formats (see “Using Attribute Mapping Expressions” on page 353). All of the variables available for text entries (see below) are also available for expressions.

• Text

The value is what you enter. This can be text only, or you can mix text with references to any of the values from the adapter, using the ${attribute} syntax.

Tip: Two other variables are also available: ${SAML_SUBJECT} and ${TargetResource}. SAML_SUBJECT is the initiating user (or other entity). TargetResource is a reference to the protected application or other resource for which the user requested SSO access; this variable is available only if specified as a query parameter for the relevant PingFederate endpoint (either as TargetResource for SAML 2.0 or TARGET for SAML 1.x—see “Application Endpoints” on page 335).

There are a variety of reasons that you might hard code a text value. For example, if your SP’s Web application provides a consumer service, you might want to supply a particular promotion code.

<table>
<thead>
<tr>
<th>Attribute Contract</th>
<th>Source</th>
<th>Value</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAML_SUBJECT</td>
<td>Adapter</td>
<td>username</td>
<td>None available</td>
</tr>
<tr>
<td>email</td>
<td>Adapter</td>
<td>email</td>
<td>None available</td>
</tr>
<tr>
<td>name</td>
<td>Adapter</td>
<td>name</td>
<td>None available</td>
</tr>
<tr>
<td>Name</td>
<td>Adapter</td>
<td>Name</td>
<td>None available</td>
</tr>
</tbody>
</table>

To reach this screen for editing:

1. Click the connection name on the Main Menu.
   - Click Manage All SP, if needed, to see a full list of connections.
2. Click Browser SSO under the SP Connection tab.
3. Click Configure Browser SSO.
4. Click Assertion Creation under the Browser SSO tab.
5. Click Configure Assertion Creation.
6. Click **IdP Adapter Mapping** on the Summary screen.
   If this step is not available, you have chosen to pass either a pseudonym or transient name identifier and not to use additional attributes (see “Choosing an Identity Mapping Method” on page 125).

7. Click the Adapter Instance Name.

8. Click **Attribute Contract Fulfillment** on the Summary screen.
   If you are using data stores for attribute mapping and this step does not appear, see “Specifying a Failsafe Attribute Source” on page 144.

**To map attributes:**

1. Choose a **Source** for each **Target** attribute (see descriptions of each **Source** under “Map each attribute to fulfill the Attribute Contract from one of these Sources:” on page 145).
2. Choose (or enter) a **Value** for each **Attribute**.
   All values must be mapped.
3. Click **Next**.

**Configuring Protocol Settings**

The **Protocol Settings** screen provides the launching point for configuring bindings, partner endpoints, and other settings needed for the selected SAML profiles (if you are using SAML 2.0—see “Choosing Profiles (SAML 2.0)” on page 121). The screen also displays configured information.

(For WS-Federation, the configuration of bindings is not applicable.)

![Protocol Settings Table]

**To configure Protocol Settings, you need to know:**

- For SSO profiles, the URL(s) of your SP’s **Assertion Consumer Service(s)**
- For SLO profiles, the URL(s) of your SP’s **Single Logout Service(s)**
- When artifact is an allowable inbound binding, the URL of your SP’s **Artifact Resolution Service(s)**
- The transport configurations (bindings) that you will use to send and receive data for SSO/SLO connections
- Digital signature policies and certification requirements to which you and your connection partner have agreed
Chapter 5
Identity Provider SSO Configuration

- XML encryption policies to which you and your connection partner have agreed

**Important:** After modifying Protocol Settings, you must click Done on the Protocol Settings screen and then Save on the Browser SSO screen.

Setting Assertion Consumer Service URLs (SAML)

At this step for SAML connections, you associate bindings to the Assertion Consumer Service (ACS) endpoint(s) where your SP will receive assertions. This configuration applies to either SSO Profile (see “Choosing Profiles (SAML 2.0)” on page 121).

**Field Descriptions**

<table>
<thead>
<tr>
<th>Field Description</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default (SAML 2.0)</td>
<td>A check in this checkbox indicates that the URL configuration in that row will be used as a default.</td>
</tr>
<tr>
<td>Index (SAML 2.0)</td>
<td>Uniquely identifies multiple ACS endpoints.</td>
</tr>
<tr>
<td>Binding</td>
<td>The method of transmission: POST or Artifact.</td>
</tr>
<tr>
<td>Endpoint URL</td>
<td>A location to which the assertion is sent, according to partner requirements.</td>
</tr>
</tbody>
</table>

**To reach this screen for editing:**

1. Click a connection name on the Main Menu.
2. Click Manage All SP, if needed, to see a full list of connections.
3. Click Browser SSO under the SP Connection tab.
4. Click Configure Browser SSO.
5. Click Protocol Settings on the Summary screen.
6. Click Assertion Consumer Service URL on the Summary screen.
To define an Endpoint URL:
1. Select the Binding your partner specifies for the Endpoint.
2. (Optional) Enter an Index value — 0 or 1, for example.
   For SAML 2.0 the specifications provide for the use of index numbers to identify multiple ACS endpoints. PingFederate supplies this number automatically; however, you can manually set the number to match your partner's configuration as needed.
3. Enter the fully qualified Endpoint URL or just a relative path if you have defined a base URL (see “General Information” on page 117).
4. For SAML 2.0 connections, if this is the default (or only) endpoint, click the checkbox under Default.
5. Click Add.

Setting a Default Target URL (SAML 1.x)
This URL is used whenever PingFederate receives an SSO request from a local application that does not include the user's target resource URL at the SP site. The URL is required regardless of whether you expect your local application(s) to specify the target—to ensure that the server functions correctly during SSO events.

Field Descriptions

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default Target URL</td>
<td>The URL of the target SP resource.</td>
</tr>
</tbody>
</table>

To reach this screen for editing:
1. Click a connection name on the Main Menu.
   Click Manage All SP, if needed, to see a full list of connections.
2. Click Browser SSO under the SP Connection tab.
3. Click Configure Browser SSO.
5. Click Configure Protocol Settings.
6. Click Default Target URL on the Summary screen.
Defining a Service URL (WS-Federation)

The Service URL is the WS-Federation endpoint of your SP partner where you send SAML assertions and SLO cleanup messages. The assertions are transmitted within an RSTR (Request for Security Token Response) message in response to a request for authentication from the SP. SLO cleanup messages are sent to WS-Federation SP partners when the IdP receives a user's SLO request. Such cleanup messages indicate that the user's local session has been terminated.

Enter the fully qualified URL or just the relative path if you have defined a base URL (see “General Information” on page 117).

You must include the initial slash if you are entering only a relative path.

Specifying SLO Service URLs (SAML 2.0)

At this step you associate bindings to the endpoints where your SP receives logout requests when SLO is initiated at your site and where you send SLO responses when you receive SLO requests from the SP.

This step applies only to SAML 2.0 connections when you select either SLO profile (see “Configuring IdP-Initiated SLO” on page 123 or “Configuring SP-Initiated SLO” on page 123).
Field Descriptions

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Binding</td>
<td>The method of transmission: POST, Artifact, Redirect, or SOAP.</td>
</tr>
<tr>
<td>Endpoint URL</td>
<td>A location to which logout messages are sent, according to SP requirements.</td>
</tr>
<tr>
<td>Response URL</td>
<td>(Optional) A location on this IdP to which logout responses are sent.</td>
</tr>
</tbody>
</table>

To reach this screen for editing:
1. Click a connection name on the Main Menu.
   - Click Manage All SP, if needed, to see a full list of connections.
2. Click Browser SSO under the SP Connection tab.
3. Click Configure Browser SSO.
5. Click Configure Protocol Settings.
6. Click SLO Service URLs on the Summary screen.

To add a URL:
1. Select the Binding type.
2. Enter the fully qualified URL (or the relative path, if you have specified a base URL—see “General Information” on page 117).
3. (Optional) Enter the Response URL.
4. Click Add.

To edit an endpoint:
1. Click Edit under Action for the endpoint.
2. Make your change and click Update.
To delete an entry:

- Click **Delete** under Action for the endpoint.

**Choosing Allowable SAML Bindings (SAML 2.0)**

At this step for SAML 2.0 connections, you select the binding(s) that your SP partner will use to send SAML authentication requests or SLO messages.

This configuration applies to SP-initiated SSO and to either SLO profile.

To reach this screen for editing:

1. Click a connection name on the Main Menu.
   - Click **Manage All SP**, if needed, to see a full list of connections.
2. Click **Browser SSO** under the SP Connection tab.
3. Click **Configure Browser SSO**.
4. Click **Protocol Settings** on the Summary screen.
5. Click **Configure Protocol Settings**.
6. Click **Allowable SAML Bindings** on the Summary screen.

**Setting an Artifact Lifetime (SAML)**

When you send an artifact to your SP's Assertion Consumer Service or SLO service (for SAML 2.0), an element in the message indicates how long it should be considered valid.
You can change the default value per your requirements, if needed. Also consider synchronizing clocks between your server and your partner’s SAML gateway server. If clocks are not synchronized, you might need to set the artifact lifetime to a higher value.

**To reach this screen for editing:**
1. Click a connection name on the Main Menu.
   - Click **Manage All SP**, if needed, to see a full list of connections.
2. Click **Browser SSO** under the SP Connection tab.
3. Click **Configure Browser SSO**.
4. Click **Protocol Settings** on the Summary screen.
5. Click **Configure Protocol Settings**.
6. Click **Artifact Lifetime** on the Summary screen.
   - This step appears only if you have selected the artifact binding for either an SSO or SLO Service (under SAML 2.0) at the SP site.

### Specifying Artifact Resolver Locations (SAML 2.0)

This endpoint or group of endpoints is where your server will send back-channel requests to resolve artifacts received from your partner. The locations are also known collectively under SAML specifications as the **Artifact Resolution Service**.

To reach this screen for editing:
1. Click a connection name on the Main Menu.
   - Click **Manage All SP**, if needed, to see a full list of connections.
2. Click **Browser SSO** under the SP Connection tab.
3. Click **Configure Browser SSO**.
4. Click **Protocol Settings** on the Summary screen.
5. Click **Configure Protocol Settings**.
6. Click **Artifact Resolver Locations** on the Summary screen.

   If this step does not appear, you do not have **Artifact** selected under **Allowable SAML Bindings**.

**To configure the Artifact Resolver Location(s):**

1. Enter a URL on the Artifact Resolver Locations screen and click **Add**.
   
   The URL must be fully qualified (defining protocol, host, and port) unless you have entered a base URL (see "General Information" on page 117).
   
   Repeat this step if your SP supports multiple services. The SAML 2.0 specifications permit multiple artifact resolution services through the use of index numbers, which PingFederate automatically supplies when you add a service. Alternatively, if needed per partner specifications, you may assign these index numbers manually.

   **Note:** When specifying multiple artifact resolution endpoints, each endpoint must share the same transport protocol. That is, if one endpoint uses HTTP, then all must use HTTP. Similarly, if one endpoint uses HTTPS, then all must use HTTPS.

2. Click **Next**.

**Configuring Signature Policy**

The Signature Policy screen provides options controlling how digital signatures are used for SSO Internet messaging. The choices made on this screen depend on your partner agreement (see “Digital Signing Policy Coordination” on page 12).

Digital signing is required for SAML Response messages sent from your site via POST (or Redirect for SAML 2.0). Optionally, SSO authentication requests from the SP (SP-initiated SSO) may also be signed to enforce security. (This option appears only for SAML 2.0 connections and only if you have enabled SP-initiated SSO using the POST or redirect bindings.)

The assertions inside SAML Responses may also be signed. When you make this choice, only the assertion portion of the Response is signed, not the complete Response. (This is the only option that appears for SAML 1.x connections.)
Managing SP Connections

Make your selection(s) and click Next, or just click Next if no additional security is required.

Configuring XML Encryption Policy (SAML 2.0)

For SAML 2.0 configurations, in addition to using signed assertions to ensure authenticity, you and your partner may also agree to encrypt all or part of an assertion to improve privacy. This feature is commonly used if the assertion might pass through an intermediary (such as a user's browser) and HTTPS is not used.

If the name identifier (or SAML_SUBJECT) of an assertion is encrypted, you and your partner may also want to encrypt the identifier in subsequent single-logout messages (if you are using an SLO profile).

Note that “The entire assertion” selection on the Encryption Policy screen includes the SAML_SUBJECT and all attributes.

To reach this screen for editing:

1. Click a connection name on the Main Menu.
   - Click Manage All SP, if needed, to see a full list of connections.
2. Click Browser SSO under the SP Connection tab.
3. Click Configure Browser SSO.
4. Click **Protocol Settings** under the Browser SSO tab.
5. Click **Configure Protocol Settings**.
6. Click **Encryption Policy** on the Summary screen.

**To define XML encryption:**

1. Choose whether you want to encrypt the entire assertion or one or more attributes.
2. If you are encrypting the name-identifier attribute, use the checkboxes near the bottom of the screen to indicate whether you will also encrypt this attribute in outbound SLO messages and/or allow its encryption for inbound messages.
3. Click **Next** or **Done**.

**To disable previously configured XML encryption selections:**

1. Select **None** and then **Done**.
2. Click **Save** on the Protocol Settings screen.

**Editing and Saving Protocol Settings**

On the Summary screen you can review or edit your Protocol Settings.

---

**Important:** When you finish editing existing settings, be sure to click **Done** on the Summary screen and then **Save** on the Protocol Settings screen. For a new connection, click **Done** and then click **Next** on the Protocol Settings screen. Save the entire connection on the Activation & Summary screen (see “Connection Activation and Summary” on page 182).

---

**To reconfigure saved settings:**

1. Click the heading over the information you want to change.
2. Click **Done** on the screen containing your change.
   
   If you need to make dependent or other changes, do so and continue by clicking **Done** until you reach the Protocol Settings screen.
3. Click **Save** on the Protocol Settings screen.

**Editing and Saving Browser SSO Settings**

On the Summary screen for Browser SSO, you can review or edit your SSO configuration.

---

**Important:** When you finish editing existing settings, be sure to click **Done** on the Summary screen and then **Save** on the Browser SSO screen. For a new connection, click **Done** and then click **Next** on the Browser SSO screen. Save the entire connection on the Activation & Summary screen (see “Connection Activation and Summary” on page 182).

---

**To reconfigure saved settings:**

1. Click the heading over the information you want to change.
2. Click **Done** on the screen containing your change. If you need to make dependent or other changes, do so and continue by clicking **Done** until you reach the Browser SSO screen.

3. Click **Save** on the Browser SSO screen.

**Configuring the Attribute Query Profile**

At the Attribute Query step you configure your connection to respond to requests for user attributes from your partner SP, if you have chosen this option (see “Choosing Connection Options” on page 115). Attribute queries are not dependent on single sign-on but may be used independently or in conjunction with Browser SSO or provisioning to provide flexibility in how a user authenticates with SP applications (see "Attribute Query and XASP" in the “Supported Standards” chapter of Getting Started).

To continue, click **Configure Attribute Query Profile**.

**Defining Retrievable Attributes**

On this screen you specify the user attributes you and your partner have agreed to allow in an attribute query transaction. Note that the SP may not necessarily request all of these attributes in each attribute-query request. Instead, the list simply limits the request to a subset of these attributes.

To add an attribute:
- Enter the attribute name in the text box and click **Add**.
To edit an attribute name:
1. Click Edit and make your change.
2. Click Update.

To delete an attribute:
► Click Delete.

Choosing a Data Store

Because no user authentication is performed in response to an attribute-query request, you cannot use attributes drawn from the user’s session (see “IdP Adapter Mapping” on page 129). Therefore, you must identify a data store that contains the attributes on your system.

Configuring Data Store Lookup

The process of configuring PingFederate to look up attributes in a data store for attribute-query responses is similar to that used for SSO Attribute Sources and User Lookup. For detailed information, see the step-by-step procedures in the sections indicated below.

If you use a JDBC data store, see:
   – “Selecting a JDBC Database Table and Columns” on page 135
   – “Configuring a Database Filter (WHERE Clause)” on page 137

If you use a LDAP data store, see:
   – “Configuring an LDAP Directory Search” on page 139
   – “Configuring an LDAP Filter” on page 140

If you use a Custom data store, see:
   – “Configuring Custom Source Filters” on page 142
   – “Selecting Custom Source Fields” on page 142

Note that the screen text may differ slightly and only one data store may be configured to supply user attributes. In addition, note that the variable
Managing SP Connections

$(SAML_SUBJECT) is available on the Database or LDAP Filter screens to retrieve the subject identifier from the Attribute Query for use in data-store query statements.

**Important:** When attribute queries are sent using XASP, use the variable $(SubjectDN) rather than $(SAML_SUBJECT) to retrieve the subject identifier. You may also use any of these DN-parsing variables: $(CN), $(OU), $(O), $(L), $(S), $(C), and $(DC).

If more than one value exists for any of the parsing variables, then they are enumerated. For example, if the Subject DN is:

cn=John Smith, ou=service, ou=employee

then you could use any of these elements in your filter qualifier:

- $(SubjectDN)=cn=John Smith, ou=service, ou=employee
- $(ou)=service
- $(ou1)=employee

For more information about XASP, see “Attribute Query and XASP” in the “Supported Standards” chapter of Getting Started.

Attribute Mapping Fulfillment

The last step in configuring an attribute source is to map values into the assertion to be sent in response to an attribute query.

You map attributes on the Attribute Mapping Fulfillment screen.

Map each attribute into the assertion from one of these Sources:

- LDAP/JDBC/Custom
  Values are returned from your attribute source. When you make this selection, the Value list is populated by the LDAP, JDBC, or Custom attributes you identified for this Attribute Source.

- Expression (when enabled)
  
  This option provides more complex mapping capabilities—for example, transforming incoming values into different formats (see “Using Attribute Mapping Expressions” on page 353). All of the variables available for text entries (see below) are also available for expressions.
Text

This can be text only, or you can mix text with references to any of the values from your user data store using this syntax:

$\{ds.attribute\}$

where attribute is any of the data store attributes you have selected.

There are a variety of reasons why you might hard code a text value. For example, if your SP's Web application provides a service based on your company's name, you might provide that attribute value as a constant.

Specifying Security Policy

This screen allows you to specify the digital signing and encryption policy to which you and your partner have agreed. These selections will trigger requirements for setting up Credentials (see “Configuring Credentials” on page 160).

To configure attribute-query security policy for this partner:
  ▶ Check or clear the check boxes and click Next or Done.

Editing and Saving Attribute Query Configurations

To reconfigure saved profiles:
1. Click the heading over the information you want to change.
2. Click Done on the screen containing your change.
   If you need to make additional changes, do so and continue by clicking Done until you reach the Attribute Query screen.
3. Click Save on the Attribute Query screen.

Configuring Credentials

The Credentials screen presents a list of possible security requirements you might need, depending on the federation protocol you are using and the choices you have made.

Your connection configuration may involve any or all of the following:

- Configuring Back-Channel Authentication
- Configuring Digital Signature Settings
Managing SP Connections

- Configuring Signature Verification Settings
- Selecting an Encryption Certificate (SAML)
- Selecting a Decryption Key (SAML)

To continue, click **Configure Credentials**.

**Configuring Back-Channel Authentication**

When you configure a profile for inbound SAML messages via the artifact binding, you must specify authentication information for outbound artifact resolution requests over SOAP to your SP’s Artifact Resolution Service.

Similarly, if you configure outbound Assertion Consumer Service or SLO Service URLs to use the artifact binding, then you must configure SOAP authentication requirements for inbound messages such as artifact resolution requests. If you configure outbound SLO Service URLs to use the SOAP binding, then you must also configure authentication requirements for outbound SOAP messages.

To reach this screen for editing:

1. Click a connection name on the Main Menu.

   Click **Manage All SP**, if needed, to see a full list of connections.
Chapter 5
Identity Provider SSO Configuration

2. Click **Credentials** under the SP Connection tab.
3. Click **Configure Credentials**.
4. Click **Back-Channel Authentication** on the Summary screen.
   
   If this step is not present, then it is not applicable to your configuration—you have not configured any profiles that use an artifact or SOAP binding or allowed artifact as an inbound SAML binding.

To configure back-channel authentication requirements for sending SOAP messages:

1. On the Back-Channel Configuration screen, click the **Configure** link to the right of the list of messages to be sent to your partner.

2. Make one or more selections on the Outbound SOAP Authentication Type screen:
   - Basic — you will enter SOAP Basic credentials on a later screen.
   - SSL Client Certificate — you will specify the certificate on a later screen.

   This option is enabled only if you have specified an endpoint that uses SSL.
   - Use Digital Signatures . . . — you will sign the message.

   You will be asked to select a signing certificate on a later screen.

   For SAML 2.0, these options may be used in any combination or independently. For SAML 1.x, you must use either Basic or SSL authentication; digital signing may be added to ensure message integrity.

   By default, PingFederate validates your partner's SSL server certificate—verifying that the certificate chain is rooted by a trusted Certificate Authority and that the hostname matches the certificate's Common Name. Clear the associated checkbox if you do not want this validation to occur.

3. Click **Next**.

4. If you chose Basic at Step 2, enter the SOAP Username and Password to use for this partner under Basic SOAP Authentication.

   You must obtain these credentials from your partner.

5. If you are using an SSL certificate, select the certificate under SSL Authentication Certificate and click **Next**.

   If you have not yet created or imported the client SSL certificate you need into PingFederate, click **Manage Certificates** (see “SSL Client Keys & Certificates” on page 91). You will need to export the certificate (only) and send it your partner.

6. On the Summary screen, click **Done**.

To configure back-channel authentication requirements for receiving SOAP messages:

1. On the Back-Channel Configuration screen, click the **Configure** link to the right of the list of messages to be received from your partner.

2. Select one or more options on the Inbound SOAP Authentication Type screen:
   - Basic — Enter the logon username and password your partner will use on the next screen.
   - SSL Certificate — Specify certificate verification information on a later screen.
• Use Digital Signatures . . . — Incoming messages must be signed.

   You will be asked to select a signature verification certificate on a later screen.

For SAML 2.0, these options may be used in any combination or independently. For SAML 1.x, you must use either Basic or SSL authentication; digital signing may be added to ensure message integrity.

3. Click Next.

4. If you chose Basic at Step 2, enter the SOAP Username and Password under Basic SOAP Authentication.

   Important: If you are configuring more than one connection that uses the artifact or SOAP profile, you must ensure that the Username is unique for each connection.

5. If you are using an SSL certificate, select Anchored or Unanchored under Certificate Verification Method.
   • Anchored — The certificate must be signed by a trusted Certificate Authority, and the CA's certificate must be imported into the PingFederate Trusted CA store (see “Trusted Certificate Authorities” on page 87).
   • Unanchored — The certificate is self-signed or you wish to trust a specified certificate.

   Note: When anchored certificates are used between partners, certificates may be changed without sending the update to your partner. If the certificate is unanchored, any changes must be promulgated.

6. Click Next.

7. If you chose anchored SSL certificate verification at Step 5, enter the Subject DN and click Next.

   Tip: If you have not yet defined the certificate in PingFederate or you do not know the DN, return to the previous screen and check Unanchored. Then click Next and click Manage Certificates on the SSL Verification Certificate screen to import the certificate, if needed, or to view its DN.

8. If you chose unanchored SSL certificate verification at Step 5, select the certificate you will use to validate the SSL connection.

   If you have not yet imported the certificate into PingFederate, click Manage Certificates.

9. Click Next.

10. On the Summary screen, click Done.
Configuring Digital Signature Settings

This step defines the private key/certificate that you will use to sign assertions and SLO messages for this SP.

**Note:** Digital signing is required for SSO assertions and SLO messages sent via POST or redirect bindings. Signing is not always required for profiles using the artifact or SOAP bindings.

The step applies to both IdP- and SP-initiated SSO and to either SLO profile (see “Choosing Profiles (SAML 2.0)” on page 121) whenever outbound POST or redirect bindings are used. The step also is required for WS-Trust STS and for SSO if you chose to sign the SAML assertion, SAML response, or artifact resolution messages (see “Configuring Back-Channel Authentication” on page 161).

To reach this screen for editing:

1. Click a connection name on the Main Menu.
   
   Click Manage All SP, if needed, to see a full list of connections.

2. Click Credentials under the SP Connection tab.

3. Click Configure Credentials.

4. Click Digital Signing Settings on the Summary screen.

   This step does not appear if a connection's configuration does not require it.

To specify a certificate:

1. Select the certificate from the drop-down list.
   
   If you have not yet created or imported your certificate into PingFederate, click Manage Certificates (see “Digital Signing and Decryption Keys & Certificates” on page 93).

2. (Optional) If you have agreed to send your public key with the SAML message, click the checkbox to include the certificate.

   **Important:** This checkbox is selected automatically for PingFederate Express connections (see “Finishing an Express Connection” on page 187).
Configuring Signature Verification Settings

Under SAML 2.0 specifications, when your site receives any SAML 2.0 messages via the POST or Redirect bindings, the messages must be digitally signed. Signing is also always required for the SAML 1.x POST binding and for WS-Federation assertions, as well as incoming SAML 1.1 or 2.0 tokens for WS-Trust STS processing.

Depending on your agreement with this SP, SSO assertions, SAML 2.0 artifacts, or SOAP messages might also require signatures.

Whenever signatures are required, PingFederate provides a choice of trust models, including an option to use anchored signature-verification certificates embedded in incoming messages (see “Trust Models” on page 12). When this option is chosen in Signature Verification Settings, you must provide the Subject DN for embedded certificates coming from this partner, and the Issuer CA certificate must be part of the PingFederate trusted store (see “Trusted Certificate Authorities” on page 87).

Alternatively, you may choose to use unanchored certificates, in which case you must import your partner’s public-key certificate during this configuration (or select it if it is already imported). To prevent any interruption of service due to an expired certificate, you can ask your partner for a new certificate in advance and import it as backup.

To continue, click Manage Signature Verification Settings.

To reach this screen for editing:
1. Click a connection name on the Main Menu.
   Click Manage All SP, if needed, to see a full list of connections.
2. Click Credentials under the SP Connection tab.
3. Click Configure Credentials.
4. Click Signature Verification Settings on the Summary screen.
   If this step does not appear, then your configuration does not require verification settings.

Choosing a Trust Model

This screen allows you to choose the Trust Model you want to use for signature verification (see “Trust Models” on page 12).
Depending on the selection, the next step in this task varies:

- For **Anchored**, the next step is to enter the Subject DN for your partner's certificate (see the next section, “Anchored Certificates: Specifying a Subject DN”).

**Important:** If you are using the Redirect binding for SLO, you cannot use anchored certificates because SAML 2.0 does not permit certificates to be included using this transport method.

- For **Unanchored**, the next step is to import your partner's certificate (see “Selecting Unanchored Certificates” on page 167).

To reach this screen for editing:

1. Click a connection name on the Main Menu.
   Click **Manage All SP**, if needed, to see a full list of connections.
2. Click **Credentials** under the SP Connection tab.
3. Click **Configure Credentials**.
4. Click **Signature Verification Settings** on the Summary screen.
   If this step does not appear, then your configuration does not require verification settings.
5. Click **Manage Signature Verification Settings**.
6. Click **Trust Model** on the Summary screen.

**Anchored Certificates: Specifying a Subject DN**

When you choose to use an anchored certificate for signature verification, incoming SAML messages must contain the partner's verification certificate (see “Trust Models” on page 12). PingFederate verifies that the Issuer CA is trusted and checks to see that the embedded certificate's Subject DN matches the one specified on this screen. If so, PingFederate uses that certificate to verify the message signature.
You can either enter the Subject DN or extract it from your partner’s certificate if the certificate is stored on an accessible file system.

**Important:** Ensure that you enter the Subject DN correctly or extract the DN from the correct partner certificate.

To extract the Subject DN from a certificate:
1. Click browse to select your SP partner’s public certificate.
2. Click **Extract**.

To reach this screen for editing:
1. Click a connection name on the Main Menu. Click **Manage All SP**, if needed, to see a full list of connections.
2. Click **Credentials** under the SP Connection tab.
3. Click **Configure Credentials**.
4. Click **Signature Verification Settings** on the Summary screen. If this step does not appear, then your configuration does not require verification settings.
5. Click **Manage Signature Verification Settings**.
6. Click **Certificate Subject DN** on the Summary screen.

**Selecting Unanchored Certificates**

On the Signature Verification Certificate screen, you identify your partner’s imported public certificate and, optionally, a secondary certificate to use when the first expires (see “Trust Models” on page 12).
To reach this screen for editing:

1. Click a connection name on the Main Menu.
   - Click **Manage All SP**, if needed, to see a full list of connections.

2. Click **Credentials** under the **SP Connection** tab.

3. Click **Configure Credentials**.

4. Click **Signature Verification Settings** on the Summary screen.
   - If this step does not appear, then your configuration does not require verification settings.

5. Click **Manage Signature Verification Settings**.

6. Click **Signature Verification Certificate** on the Summary screen.

To specify a verification certificate:

1. Select the certificate from the drop-down list.
   - If you have not yet imported the certificate into PingFederate, click **Manage Certificates**.

2. (Optional) Select a Secondary certificate for backup.
   - Use this field if your partner has sent you a new certificate to replace one that is ready to expire. The server will automatically verify against the secondary certificate when the primary one expires.

**Selecting an Encryption Certificate (SAML)**

To enable XML encryption of all or part of an SSO assertion, you must identify the encryption certificate you will use (see “Configuring XML Encryption Policy (SAML 2.0)” on page 155).

You must also select a certificate if your requirements include encrypting an assertion in response to an attribute query (see “Specifying Security Policy” on page 160).
Managing SP Connections

To reach this screen for editing:

1. Click a connection name on the Main Menu.
   - Click Manage All SP, if needed, to see a full list of connections.
2. Click Credentials under the SP Connection tab.
3. Click Configure Credentials.
4. Click Select XML Encryption Certificate.

If this step is not present, you have chosen not to encrypt the assertion or the SAML_SUBJECT (see “Configuring XML Encryption Policy (SAML 2.0)” on page 155).

To identify the encryption certificate:

1. (Optional) Change the default settings under Block Encryption Algorithm and/or Key Transport Algorithm.

   Due to import control restrictions, the standard JRE distribution supports strong but not unlimited encryption. To use the strongest AES encryption, when permissible, download and install the appropriate version of “Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files” from the Sun download Web site (http://java.sun.com/javase/downloads).

   For more information about XML block encryption and key transport algorithms, see the “XML Encryption Syntax and ProcessingW3C Recommendation” at http://www.w3.org/TR/xmlenc-core/.

2. From the drop-down list, select the applicable certificate and click Next.
   - If the certificate is not in the list, click Manage Certificates to import it.

   **Note:** If you have already imported a signature verification certificate for this partner, you can reuse it for XML encryption as long as it is an RSA certificate.

Selecting a Decryption Key (SAML)

If SAML_SUBJECT is encrypted, either by itself or as part of a whole assertion, then all references to this name identifier in SLO requests from your partner may also be encrypted (if the connection uses SP-initiated SLO under SAML 2.0).
To enable XML encryption, you must identify a certificate for PingFederate to use to decrypt incoming SLO messages.

To reach this screen for editing:
1. Click a connection name on the Main Menu.
   - Click Manage All SP, if needed, to see a full list of connections.
2. Click Credentials under the SP Connection tab.
3. Click Configure Credentials.
4. Click Select XML Decryption Key.
   - If this step is not present, you have chosen not to encrypt the assertion or the SAML_SUBJECT attribute (see “Configuring XML Encryption Policy (SAML 2.0)” on page 155).

To identify the decryption key:
- From the drop-down list, select the applicable certificate and click Next.
  - If the certificate is not in the list, click Manage Certificates to import it (see “Digital Signing and Decryption Keys & Certificates” on page 93).

**Note:** If you have imported a certificate to use for digital signing, you can reuse it for XML decryption as long as it is an RSA certificate.

**Editing and Saving Credential Configurations**

From the Summary screen you can review or edit your credentials configuration.

**Important:** When you finish editing existing settings, you must click Done on the Summary screen and then Save on the Credentials screen. For a new connection, click Done and then click Next on the Credentials screen. Save the entire connection on the Activation screen (see “Connection Activation and Summary” next).

**Configuring SaaS Provisioning**

PingFederate's SaaS Provisioning (available separately) allows an IdP to create and maintain user accounts at selected service-provider sites in order to streamline SSO
Managing SP Connections

(see “SaaS Provisioning” on page 18). Detailed configuration of this feature begins on the SaaS Provisioning screen.

**Note:** This configuration task is presented in the administrative console only when SaaS Provisioning has been enabled and selected as an option for the current connection (see “Choosing Connection Options” on page 115). For new connections, you must first configure Browser SSO and Credentials settings (see “Identity Provider SSO Configuration” on page 101).

![SAML20](image)

To continue, click **Configure Provisioning**.

### Defining a Provisioning Target

Information on the Target screen indicates what account name and password PingFederate will use for authentication to the provisioning API for either Google Apps or salesforce.com. To use this screen, you must have a registered domain with the SaaS provider and a valid account and password that can be used to access the provider's public Web-service endpoint for provisioning.

For Salesforce, the screen also provides a choice of environments (production or testing).
Chapter 5
Identity Provider SSO Configuration

To configure the Target:
1. Enter a valid Admin Email (for Google) or Admin Username (for Salesforce).
2. Enter the Admin Password.
3. For Salesforce channels, select the Environment.
4. Click Next.

Note: When you first enter or change credentials and click Next, PingFederate immediately tests connectivity to the target.

Managing SaaS Channels

A provisioning channel is a mapping configuration between user attributes contained in a source user store and attributes supported or required by the targeted software-service application. You can have multiple channels to the same target as needed—for example, if your organization has separate LDAP stores (or different nodes in the same store) for various user groups needing SSO access and provisioning to the same SaaS domain.

Tip: There can be only one target SaaS domain for a connection ("Defining a Provisioning Target" on page 171). If your organization subscribes to multiple SaaS domains for which you need provisioning and SSO support, you will need a separate SP connection for each domain.
Channels are created and managed from the Manage Channels screen.

To access the Manage Channels screen:
1. In the task headings for a connection, click SaaS Provisioning. If this task is not present, provisioning is not enabled (see “Choosing Roles and Protocols” on page 64).
2. On the SaaS Provisioning screen, click Configure Provisioning.

To configure a new channel:
- Click Create and follow the configuration steps.

Tip: If you are creating a second channel to a SaaS vendor, you may wish to copy an existing channel and make necessary changes.

To copy a channel:
1. Click Copy under Action for the channel you want to copy.
2. Enter new General Info for the channel (see “Specifying Channel Information” on page 173).
3. Make any further changes needed for the new channel.

To edit a channel:
- Click the Channel Name link.

To delete a channel:
1. Under Action, click Delete for the channel. (To undo the deletion, click Undelete.)

Note: The Delete function is not available if the channel is active (see “Channel Activation and Summary” on page 181).

2. To confirm the deletion, click Done and then Save on the Configure SaaS Provisioning screen.

Specifying Channel Information

On the Channel Info screen, specify a unique identifier for the channel.
Identifying the Source Data Store

PingFederate fully supports Active Directory and the Sun Directory Server as source user repositories for SaaS Provisioning. However, you can use other types of LDAP servers, either identifying them as Generic or registering them with PingFederate (see “Configuring an LDAP Connection” on page 77).

Information from your user data store is used to supply mapped values for each user attribute required by the SaaS provider (see “Mapping Attributes” on page 178).

Tip: Adjust the Max Threads setting as needed to optimize data-transfer performance, particularly if large numbers of records need to be provisioned at the target site.

On the Source screen, choose the LDAP store to use for this channel.

Note: If the correct data store is not listed, then it has not yet been identified to PingFederate; click Manage Data Stores to set a connection to user store (see “Configuring an LDAP Connection” on page 77).

Modifying Source Settings

The Source Settings screen shows the default configuration of the data store selected on the previous screen, including settings used by the PingFederate
provisioner to determine when user information has been added, changed, or removed.

If you are using the Sun Directory Server or Active Directory, in most cases no changes are needed on this screen unless your data store uses a customized schema.

If you are using a different LDAP directory, you must supply the required information on this screen unless you have defined a template for the data store (see “Defining an LDAP Type” on page 79).
## Field Descriptions

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entry GUID Attribute</td>
<td>The name of the attribute in the data store representing the user’s Globally Unique Identifier.</td>
</tr>
<tr>
<td>GUID Type</td>
<td>Indicates whether the GUID is stored in binary or text format. Active Directory is always binary. Other LDAP stores most often use text.</td>
</tr>
<tr>
<td>Member of Group Attribute</td>
<td>A multi-value user attribute containing theDNs of the groups to which an entry belongs. This attribute does not apply to some LDAP servers,</td>
</tr>
<tr>
<td></td>
<td>including the Sun Directory Server. The attribute below is used instead. Active Directory uses both values to provide a two-way mapping between</td>
</tr>
<tr>
<td></td>
<td>User and Group objects.</td>
</tr>
<tr>
<td>Group Member Attribute</td>
<td>The name of a multi-value group attribute used to track membership in the group using either DN or GUID values.</td>
</tr>
<tr>
<td>User Object Class</td>
<td>The LDAP object class to which user entries belong, used to restrict search results to user entries only.</td>
</tr>
<tr>
<td>Changed Users Algorithm</td>
<td>The method by which PingFederate determines if user records have been updated or new records added, thus requiring provisioning updates at</td>
</tr>
<tr>
<td></td>
<td>the target SaaS site. The three choices are:</td>
</tr>
<tr>
<td></td>
<td><strong>Active Directory USN</strong> – For Active Directory only, this algorithm queries for update sequence numbers on user records that are larger</td>
</tr>
<tr>
<td></td>
<td>than the last time records were checked.</td>
</tr>
<tr>
<td></td>
<td><strong>Timestamp</strong> – Queries for timestamps on user records that are not older than the last time records were checked. This check is more</td>
</tr>
<tr>
<td></td>
<td>efficient from the point of view of the PingFederate provisioner but can be more time consuming on the LDAP side, particularly with the Sun</td>
</tr>
<tr>
<td></td>
<td>Directory Server.</td>
</tr>
<tr>
<td></td>
<td><strong>Timestamp No Negation</strong> – Queries for timestamps on user records that are newer than the last time records were checked. This algorithm</td>
</tr>
<tr>
<td></td>
<td>is recommended for the Sun Directory Server.</td>
</tr>
<tr>
<td>USN Attribute</td>
<td>The name of the attribute used to store the update sequence number—applicable when the Active Directory algorithm is chosen above.</td>
</tr>
<tr>
<td>Timestamp Attribute</td>
<td>The name of the attribute used to store the timestamp on user records.</td>
</tr>
<tr>
<td>Account Status Attribute</td>
<td>The name of the attribute in which the user’s account status (active or inactive) is stored.</td>
</tr>
</tbody>
</table>
Specifying a Source Location

Indicate on the Source Location screen where PingFederate should look for user records in the data store.

After specifying the required Base DN, you have two options for indicating which users are to be provisioned: the search can be based either on a specified LDAP filter or on group membership. (If needed for special situations, both methods can be used.)

Field Descriptions

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Base DN</td>
<td>The base distinguished name of the tree structure where user records are stored. PingFederate looks only at this node level or below it for user accounts that need to be provisioned.</td>
</tr>
</tbody>
</table>
To specify a location:
1. Enter the User Base DN.
2. Enter either a Filter, a Group DN (or both if needed).

Mapping Attributes

The Attribute Mapping screen provides a means of managing how attributes from your user store are mapped to the provisioning fields supported for your organization's SaaS-customer account.

**Tip:** PingFederate automatically retrieves from the SaaS vendor the Field Names shown on this screen, but only on the first pass through the screen flow. If you are using this screen to modify an existing mapping configuration, click **Refresh Fields** near the bottom of the screen to synchronize the list with the target data store if needed.
For each field, the screen provides a means of adding or modifying the mapping details (see the next section, "Specifying Mapping Details").

**Note:** All required attributes listed in the Field column, indicated with asterisks, must be mapped. Click View Partner Field Specifications near the bottom-left of the screen for a summary of requirements for all fields specified for the target SaaS partner.

For some fields, PingFederate preselects LDAP attributes commonly used to store the required values.

**To configure attribute mapping:**

1. Click Edit under Action for a field.
2. On the Specify Attribute Mapping screen, provide mapping details.
   (For more information, see the next section, “Specifying Mapping Details”.)
3. Repeat for each attribute shown in the Field Name column as needed.
Specifying Mapping Details

On this screen, you define specific mapping information for each field required for provisioning (and for any optional fields, as needed).

**Tip:** If you need to map more than one attribute from your data store into a single field at the target location, then you must use an OGNL expression to indicate how the attribute values are to be combined. The use of OGNL expressions may not be enabled for your PingFederate installation (see “Using Attribute Mapping Expressions” on page 353).

**Caution:** If end-users at your site are permitted to edit some of their own attributes directly in the LDAP store, ensure that the attributes are restricted and do not include any attributes needed by the SaaS provider to grant permissions.

To define mapping information for an attribute:

1. Select the Root Object Class containing a user-store attribute that you want to map to the SaaS attribute shown under Field.

**Note:** For some required fields, you may not need to map specific user attributes. If so, supply a Default Value instead—skip this step and go to Step 5. You can also do both as needed: specify LDAP attributes as well as a Default Value.

2. Under Attribute, select an attribute from the class.

3. Click **Add Attribute**.
4. Repeat the steps above to add additional applicable attributes, as needed, to use in a mapping expression.

5. Under Value Definition, enter or select a Default Value (optional, if one or more attributes is specified above).
   A drop-down list is shown for this field if the SaaS vendor requires a choice among specified values. When an expression is also supplied, the default value is sent during provisioning if an error occurs evaluating the expression.

6. If more than one attribute is used for mapping, enter an Expression.

   **Tip:** Click Test to validate the expression.

   For information about the expression language supported by PingFederate, OGNL, see “Using Attribute Mapping Expressions” on page 353.

   **Important:** The use of OGNL expressions may not be enabled for your PingFederate installation (see “Enabling/Disabling Expressions” on page 354).

7. (Optional) Select one or more processing Options, as defined below:
   - Create Only – The field is provisioned only once and not subsequently updated.
   - Trim – Removes any white space from the attribute value(s).
   - Upper Case/Lower Case/None – Transforms the attribute value(s) to the case indicated, unless None is selected (the default).

**Channel Activation and Summary**

When you finish setting up a channel, you may choose to activate it immediately; or you can return to the Activation & Summary screen and activate the channel when needed. Note that the overall SP connection for the SaaS partner also must be active for any provisioning channels to be enabled (see “Connection Activation and Summary” on page 182).

**Caution:** When a connection containing a newly activated channel is itself activated, initial provisioning occurs as soon as the synchronization-frequency time period expires (see “Configuring SaaS Provisioning Settings” on page 69). The default is 60 seconds. Initial provisioning can consume considerable processing time, depending on the amount of data that needs to be transmitted; administrators may wish to plan accordingly.

**Important:** Regardless of whether you choose to activate a new channel immediately or later, if you want to save the channel configuration, click Done on the Summary screen and then Save on the connection Activation and & Summary screen. (For a new channel in an existing connection, click Save on the SaaS Provisioning screen.)

You can deactivate a channel at any time (for maintenance, for example). When a channel is inactive, SSO/SLO transactions can still occur (if the connection is active), but provisioning is suspended.
To change a channel status:

▶ Select either Active or Inactive and then click Done.

To modify a channel setting:

▶ If you know which step needs to be modified, click its link under the SaaS Channels tab.

If you do not know where to change a setting, locate the currently configured setting under one of the summary headings and then click the subheading above the information.

Connection Activation and Summary

When you finish setting up a connection, you may choose to activate it immediately.

---

**Important:** Regardless of whether you choose to activate a new connection now or later, you must click Save on the Summary screen for a new connection if you want to keep the configuration.

---

You can deactivate a connection at any time (for maintenance, for example). When a connection is inactive, all SSO or SLO transactions to or from this partner are disabled, as well as access to the WS-Trust STS for Web Service Clients associated with this connection.

---

**Tip:** The SSO Application Endpoint near the top of the Summary screen is an example URL that webmasters or Web application developers at your site might use to invoke SSO for the connection. For details about SSO and other server endpoints, including optional query parameters, see “Application Endpoints” on page 335.

---

To change a Connection Status:

▶ Select either Active or Inactive and then click Save.

To modify a connection setting:

1. If you know which step needs to be modified, click its link under the SP Connection tab.

If you do not know where to change the setting, locate the currently configured data under one of the summary headings and then click the subheading above the data.

2. Change the information on the step screen and click Save, if available.

If Save is not available, you are in the middle of a task (see “About Tasks and Steps” in the “Console Navigation” chapter of Getting Started); click Next or Done until you reach a screen containing a Save button. Then click Save and continue as needed until you return to the Main Menu.

If your modification requires related configuration changes, PingFederate provides error messages indicating the necessary steps and then guides you to the related screens (unless you click Cancel).

---

**Important:** Be sure to click Save whenever that button appears, if you want to keep your changes.
Defining SP Affiliations

An SP affiliation is a SAML 2.0 specification that permits a group of service providers to make use of the same persistent name identifier for account linking (see “Account Linking” on page 5).

This may be of use when multiple service providers share a business relationship in which users need services from each affiliated provider. By agreement among the affiliation members, the same pseudonym can be used to populate the SAML_SUBJECT of assertions sent to all of the SP partners contained in this affiliation.

Important: Each connection in the affiliation must be configured to use the same IdP adapter instance for generating account links (see “IdP Adapter Mapping” on page 129).

You can create or modify an SP affiliation from the Main Menu or from a list of affiliations (click Manage All Affiliations).

To create an SP affiliation:

- Click Create New under SP Affiliations on the Main Menu.
  Or:
- Click Manage All Affiliations and then click Create Affiliation on the Select An Affiliation screen.

To delete an affiliation:

1. Click Manage All Affiliations under SP Affiliations on the Main Menu.
2. Click Delete under Action for the affiliation you want to delete.
3. Click Save to confirm the deletion (or click undelete).

To view or modify an affiliation:

- Click the affiliation name, or click Manage All Affiliations if the ID does not appear.

Using the Manage Affiliations Screen

You can manage SP affiliations on this screen.
Chapter 5
Identity Provider SSO Configuration

To reach this screen for editing:
► Click Manage All Affiliations under SP Affiliations on the Main Menu.

To begin creating a new affiliation:
► Click Create Affiliation (see the next sections for more information).

To delete an affiliation:
1. Click Delete under Action for the affiliation you want to delete.
2. Click Save to confirm the deletion (or click undelete).

To view or modify an affiliation:
► Click the affiliation ID.

Importing Affiliation Metadata

An IdP may send a metadata file containing information that automatically specifies members of an SP affiliation for use in PingFederate.
► If you do not have a metadata file, click Next.

To import metadata:
1. Click Browse to locate and import the file and then click Next.
2. Review the information on the Create Affiliation page (see the next section).
3. Click Save on the Summary screen.

Entering Affiliation Information

Enter or modify basic information about an affiliation on the Affiliation General Info screen.

If you imported a metadata file, this information is already supplied. However, you may change the Affiliation ID or select a different Affiliation Owner, if required.
Defining SP Affiliations

Field Descriptions

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Affiliation ID</td>
<td>A unique identifier for this affiliation. This value serves as the Name ID qualifier for SAML assertions sent to affiliated SP partners.</td>
</tr>
<tr>
<td>Affiliation Owner</td>
<td>Any SAML 2.0 SP connection may serve as the Owner.</td>
</tr>
</tbody>
</table>

Managing Affiliation Membership

On the Affiliation Membership screen, you create and manage a list of SP connections to be included in the affiliation.

If you imported a metadata file, this information is already supplied. However, you may add or remove connections from the affiliation.

► To add an SP partner connection to the affiliation, select the connection from the drop-down list and click **Add**.

重要: 每个连接在联盟中都必须配置为使用相同的IdP适配器实例来生成账户链接（参见“IdP Adapter Mapping”在第129页）。

► To remove a member of the affiliation, click **Delete** under **Action** for the connection and click **Save**.
Activating and Editing the Affiliation

From the Affiliation Management Summary screen you can activate or deactivate an SP affiliation. You also save new affiliations on this screen, or you can click heading links to go back and modify information.

**To change an Affiliation Status:**

- Select either Active or Inactive and then click **Save**.

**Important:** Be sure to click **Save**. Otherwise, the status will not be changed.

**To edit a connection:**

1. Click the heading above the information you want to modify.
2. Make your change and click **Save**.

Using the Express Connection Template

When creating a new connection, an administrator can choose to load the PingFederate Express template (see “About PingFederate Express” on page 14). This template configures most browser-based SSO settings automatically for an Express-enabled SP partner, who can then use the exported metadata for rapid end-to-end SSO configuration and deployment.

This section provides step-by-step instructions for filling in the minimal site-specific information needed after you select the template, and then for exporting the metadata.

**Important:** The PingFederate Express SP software supports a basic, lightweight configuration needed for simplified end-to-end deployment of SAML 2.0 IdP- and SP-initiated SSO, exclusively. While some minor settings may be modified for an Express SP connection (as indicated in the following section), an IdP administrator cannot extend the configuration to include broader use cases such as different protocols or bindings, WS-Trust STS, provisioning, or SLO. Although these and other features are not explicitly disabled, the connection metadata cannot be exported if any unsupported extensions are included.
Finishing an Express Connection

After selecting the PingFederate Express template, two site-specific setups must be completed, including at least one IdP adapter instance and the selection of a signing certificate for SAML assertions.

**Tip:** The administrative console guides you to required configuration steps automatically by displaying prompts (in the form of error messages) at entry points for the task flows (see “About Tasks and Steps” in the “Console Navigation” chapter of Getting Started).

Once the connection is saved, you can export its metadata into a configuration ZIP file for your partner to use when setting up PingFederate Express (see “About PingFederate Express” on page 14).

**To complete a PingFederate Express SSO Connection:**

1. After selecting the PingFederate Express template on the Connection Template screen, enter the Target Application URL (see “Choosing a Connection Template” on page 114 or the online Help page).

2. (Optional) On the General Info screen, enter further identifying information about this SP partner.

   You may also use a Virtual Server ID or adjust the Logging Mode. (See “Federation Server Identification” on page 20 and “Transaction Logging Modes” on page 27, respectively; or see the online Help pages.)

3. On the Browser SSO screen, click **Configure Browser SSO**.

4. On the Assertion Creation screen, click **Configure Assertion Creation**.
Chapter 5
Identity Provider SSO Configuration

5. (Optional) If you wish to extend the **Attribute Contract**, click that step. For more information, see “Creating an Attribute Contract” on page 127 or the online Help page.

6. On the IdP Adapter Mapping screen, click **Map New Adapter Instance**.

7. Follow procedures to map an Adapter Instance (see “IdP Adapter Mapping” on page 129 or the online Help pages).

8. Map additional Adapter Instances as needed.

9. Click **Done** on the IdP Adapter Mapping screen and **Done** again on the Assertion Creation screen.

10. On the Browser SSO screen, click **Next** to reach the Credentials screen.

11. Click **Configure Credentials**.

12. On the **Digital Signature Settings** screen, select a signing certificate to use for this partner.

   For more information, see “Configuring Digital Signature Settings” on page 164 or the online Help page.

---

**Important:** Retain the check-box selection to include the certificate (public key).
13. Click **Done** and then **Next** to reach the Activation & Summary screen.

14. Activate (optional) and **Save** the connection.

   For more information, see “Connection Activation and Summary” on page 182 or the online Help page.

**To export metadata for your partner’s configuration:**

**Either:**

- On the Main Menu under Administrative Functions, click **Metadata Export** and follow the steps, selecting the desired Express connection on the Connection Metadata screen.

  See “Exporting Metadata” on page 27 or the online Help pages.

**Or:**

1. On the Main Menu under SP Connections, click **Manage All SP**.
2. On the Manage Connections screen, click **Export Metadata** under Action for the Express connection.
3. Save the ZIP file.

Your Express partner needs the exported ZIP file to configure the SP end of the connection.

---

**Configuring SP Auto-Connect**

When your SP partner is also using PingFederate 5 or higher (or is otherwise able to provide interoperable SAML 2.0 metadata via HTTP on demand), you may choose to use Auto-Connect for that partner (see “Using Auto-Connect” on page 15). This configuration can be shared among an unlimited number of SAML 2.0 partners.

**Tip:** If you choose a CA-signed certificate, the SP Express installation will automatically use an anchored-certificate trust model (see “Trust Models” on page 12), and the CA certificate will be sent along with the verification (public-key) certificate in the exported metadata. If you choose a self-signed certificate, the SP installation will use an unanchored-certificate trust model, which means you and the SP will need to reconfigure when the certificate expires or is updated.

**Note:** You enable the SAML 2.0 Auto-Connect profile under System Settings (see “Choosing Roles and Protocols” on page 64).

Once Auto-Connect is enabled on your PingFederate server, you complete the configuration from the Main Menu under My IdP Configuration. This configuration entails:

- Setting up a common connection for all Auto-Connect partners
- Establishing a list of SP partner domains authorized to use the connection

**Initial Setup**

The basic configuration for SP Auto-Connect requires only:

- Defining a period of validity for **assertions** (assertion lifetime)
- Choosing a signing certificate for assertions and other SAML messages
Configuring assertion-creation information

All other partner-connection specifications are handled automatically at runtime.

Setting an Assertion Lifetime

Identity-federation standards require a window of time during which an assertion is considered valid. Each assertion has a time-stamp XML element as well as elements indicating the allowable lifetime of the assertion (in minutes) before and after the assertion time stamp.

Field Descriptions

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minutes Before</td>
<td>The amount of time before the assertion was issued during which it is to be considered valid.</td>
</tr>
<tr>
<td>Minutes After</td>
<td>The amount of time after the assertion was issued during which it is to be considered valid.</td>
</tr>
</tbody>
</table>

To change the default times:

► (Optional) Edit the desired setting(s) and click Next or Save.

Choosing a Signing Certificate

For Auto-Connect runtime processing, assertions and SLO messages must be signed, since they are sent over either the POST or redirect bindings (see “SAML 2.0 Profiles” in the “Supported Standards” chapter of Getting Started).

Note: The signing certificate is embedded in your server’s Auto-Connect metadata (see “Using Auto-Connect” on page 15); there is no need to exchange certificates with your partners.
You can use the same certificate used for signing metadata (see “Configuring Auto-Connect Metadata Signing” on page 70). If you use a different certificate, ensure that it meets Auto-Connect validation requirements (see “Auto-Connect Security Model” on page 17).

### Configuring Assertion Creation

Configuring assertion creation for Auto-Connect is similar to configuring the same settings for regular partner connections.

![Configuration Screen](image)

- Click **Configure Assertion Creation** to continue.

  For configuration information, refer to sections under “Assertion Creation” on page 124.

### Auto-Connect Activation and Summary

When you finish configuring your SP Auto-Connect initial setup, you may choose to activate the common connection immediately on the Activation & Summary screen. (No runtime processing occurs until your partner’s Auto-Connect gateway is also established and a user initiates an SSO or SLO event.)

**Important:** Regardless of whether you choose to activate a newly configured connection now or later, you must click **Save** on the Activation & Summary screen if you want to keep the configuration.
You can deactivate the connection at any time (for maintenance, for example). While a connection is inactive, all SSO or SLO transactions to or from Auto-Connect partners are disabled.

**To change a Connection Status:**

- Select Active or Inactive and then click Save.

**To modify a setting:**

1. Locate the currently configured setting under one of the summary headings and then click the subheading above the data.

   **Note:** Changes made to Auto-Connect settings will be out of sync, temporarily, with metadata caches that any currently active partners might be using. If your connection is in production, you might wish to lower your server’s metadata lifetime in advance of making configuration changes (see “Configuring Auto-Connect Metadata Lifetime” on page 71).

2. Change the information and click Save, if available.

   If Save is not available, additional, dependent changes are required; click Next or Done until you reach a screen containing a Save button. Then click Save and continue as needed until you return to the Main Menu.

### Specifying Allowed SP Domains

This screen provides PingFederate with a list of trusted domain names of your Auto-Connect partners.

Normally, when PingFederate receives an authentication request from a domain in this list, the runtime engine completes the connection automatically using metadata obtained from a standard, public location—http://saml.<domain_name>. (See “Using Auto-Connect” on page 15.) Alternatively, if an Auto-Connect partner elects not to use the standard location, you can supply the applicable URL.
In an SP role, you use the PingFederate administrative console to configure local application-integration information and to manage connections to your IdP-partner sites. You must configure Server Settings from the Main Menu to establish your site as an SP before configuring connections to IdPs (see “Choosing Roles and Protocols” on page 64).

Note that only one connection is needed per partner, even if you are integrating more than one Web application. You can configure more than one connection, however, if your partner supports multiple protocols, or supports multiple federation IDs for the same protocol (see “Federation Server Identification” on page 20).

Under some conditions, you can enable SSO for an unlimited number of partners at once by configuring a single, common connection (see “Using Auto-Connect” on page 15).

This chapter covers the following major topics:

- “Application Integration Settings” on page 194
- “Federation Settings” on page 203
- “Managing IdP Connections” on page 206
- “Configuring IdP Auto-Connect” on page 265
Chapter 6
Service Provider SSO Configuration

Application Integration Settings

The integration of local applications with PingFederate is the essential “last-mile” configuration that allows end-users at your IdP partner’s Web site to access your protected resources. This process is facilitated through the use of application-integration kits and a robust Software Development Kit (see “SSO Integration Kits and Adapters” on page 4).

Under Application Integration Settings on the Main Menu, you configure the SP Adapters that PingFederate uses to create user sessions that allow SSO access to your protected resources. You can also set Default URLs to which users may be directed during SSO or SLO, and you can look up system endpoints that application developers at your site need to access PingFederate’s SSO/SLO services.

**Note:** If your PingFederate configuration enables the WS-Trust STS, the selections under Application Integration Settings also include a link for configuring plug-in Token Generators (see “Configuring Token Generators” on page 300).

Configuring SP Adapters

SP adapters are used to create a local-application session for a user in order to provide SSO access to your application(s) or other protected resources (see “SSO Integration Kits and Adapters” on page 4). You can configure multiple instances of adapters (based on one or more adapters) to accommodate the varying needs of your IdP partners.

**Note:** If you are configuring either the OpenToken or the LDAP Adapter, see “Configuring the IdP OpenToken Adapter” on page 320 or “Configuring the IdP LDAP Adapter” on page 329, respectively.

If you configure more than one adapter instance, then you must map a target URL to at least one instance (see “Mapping URLs to Adapter Instances” on page 198).

SP adapter setup is available only if your server is configured as an SP (see “Choosing Roles and Protocols” on page 64).

**Important:** If you install a new version of an adapter JAR file after setting up connections to instances of that adapter, you might need to reconfigure those connections. To find out, click each connection that uses the adapter (see “Accessing Connections” on page 207). Errors indicating reconfiguration points may be presented.
You reach this screen by clicking **Adapters** under Application Integration Settings in My SP Configuration.

**To create a new adapter instance:**
- Click **Create New Instance**.
  
  See the next section.

**To edit an adapter instance:**
- Click the **Instance Name** link.

**To delete an adapter instance:**
1. Click **Delete** next to the **Instance Name** on the Manage SP Adapter Instances screen. (To undo the deletion, click **Undelete**.)

   **Note:** This option is available only if the adapter instance is not in use for a connection.

2. Click **Save** to confirm the deletion.

**Creating an Adapter Instance**

On the **Type** screen, you begin creating an instance of an adapter that PingFederate will use for creating security sessions for your applications.
Chapter 6
Service Provider SSO Configuration

Field Descriptions

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instance Name</td>
<td>A descriptive name for the adapter instance—for example the target application or group of applications.</td>
</tr>
<tr>
<td>Instance ID</td>
<td>An internal identifier for the adapter instance. Must be alphanumeric with no spaces.</td>
</tr>
<tr>
<td>Type</td>
<td>A list of previously deployed session creation adapter types that are available to create an adapter instance for the server. You can configure any number of instances for a server acting as an SP.</td>
</tr>
</tbody>
</table>

To reach this screen:
1. Click Adapters on the Main Menu.
2. Click Create New Instance on the Manage SP Adapter Instances screen.

To define an adapter instance:
1. Enter the Instance Name and Instance Id on the Type screen.
2. Select the Type from the drop-down menu.
   If the adapter you need is not listed, click Visit PingIdentity.com for additional types to see if a suitable adapter is available from the PingFederate download site. You can also create your own adapter (see “SSO Integration Kits and Adapters” on page 4).
3. Click Next and enter information on subsequent screens for this adapter setup, as indicated in the following sections.

   Tip: The setup steps and information needed vary with the adapters deployed on your server (see “SSO Integration Kits and Adapters” on page 4). For information about configuring the adapters packaged with PingFederate, see “OpenToken Adapter Configuration” on page 319 or “LDAP Adapter Configuration” on page 327.

4. Click Done on the Summary screen.
5. Click Save on the Manage SP Adapter Instances screen.

To view or modify adapter settings:
▶ Click the Instance Name.

To delete an adapter instance:
1. Click Delete next to the Instance Name on the Manage Adapter Instances screen. (To undo the deletion, click Undelete.)

   Note: This option is available only if the adapter instance is not in use for any connection.

2. Click Save to confirm the deletion.
Configuring an Adapter Instance

Configuration parameters on the SP Adapter Instance screen vary according to the adapter you choose. These options are controlled by the adapter software (see “SSO Integration Kits and Adapters” on page 4).

► For information about configuring the OpenToken Adapter, see “Configuring the SP OpenToken Adapter” on page 323.
► For information about configuring the LDAP Authentication Service, see “Configuring the SP LDAP Adapter” on page 332.

Invoking Adapter Actions

Adapters can be written to perform configuration assistance or validation actions—for example, testing a connection to an active directory. Actions may also include generation of parameters that might need to be set manually in a configuration file.

► For information about actions available using the OpenToken Adapter, see “OpenToken Adapter Configuration” on page 319.
► For information about actions available using the LDAP Authentication Service, see “LDAP Adapter Configuration” on page 327.

To reach this screen for editing:

1. On the Main Menu under Application Integration Settings for My SP Configuration, click Adapters.
2. Click an Instance Name.
3. Click Actions (if available).

To generate a properties list:

► Click Download under Action Invocation Link.

Extending an Adapter Contract

Adapters may be written with an option allowing administrators to add to the attributes required for creating usable sessions. This feature might be needed, for example, by a legacy application that requires different authentication than other applications under the same enterprise identity-management system.
To reach this screen for editing:

1. On the Main Menu under Application Integration Settings for My SP Configuration, click Adapters.
2. Click an Instance Name.
3. Click Extended Contract (if available).

To add an attribute:

1. Enter the attribute name in the text box and click Add.
2. Click Done then click Save on the Manage SP Adapter Instances page.

Editing and Saving SP Adapter Instances

From the Adapter Instance Summary screen, you can reach adapter settings for editing.

To edit the configuration:

1. Click the heading above the information you want to change.
2. Click Save on the configuration page and on the Manage SP Adapter Instances screen.

To save an adapter instance:

1. Click Done on the Summary screen.
2. Click Save on the Manage SP Adapter Instances screen.

**Note:** If this is the second adapter instance you have configured, then Save is not yet available; you must choose whether to map the new adapter instance to an application or resource URL. In this case, click Next to continue (see “Mapping URLs to Adapter Instances” next).

Mapping URLs to Adapter Instances

When you configure more than one SP adapter instance, you must map target URLs to at least one adapter instance. Mapping enables you to direct inbound SAML messages to the appropriate application.
For example, this mapping configuration may be necessary in an IdP-initiated SSO scenario that connects to multiple applications at your site. For transactions initiated at your site, this mapping is needed for default situations, in cases where the target and adapter instance are not specified when the SSO/SLO is started (see “SP Endpoints” on page 338). (When this information is provided with the SP request, the mapping table is ignored.)

This screen is available only if your server is configured as an SP and if you are using more than one adapter instance, or if you have previously mapped an adapter instance on this screen.

Field Descriptions

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>URL</td>
<td>The target URLs that align with your configured adapter instances. The URLs instruct the PingFederate SP server to route session-creation processing through an adapter instance. If the URL in the incoming request is not matched by the first entry in this table, subsequent entries are tried until a match is found.</td>
</tr>
<tr>
<td>Adapter Instance (drop-down menu)</td>
<td>A selection of configured SP adapter instances.</td>
</tr>
</tbody>
</table>

The order of mapping is significant in that the first matching mapping, from top to bottom, determines which adapter instance receives the SAML message. For example, if two URLs are mapped in the following order:

a. http://yourapp.com/subapp/* Adapter 1  

The URL http://yourapp.com/subapp/start will map to Adapter 1 because it matches mapping a. If the order of the mappings were reversed, http://yourapp.com/subapp/start would map to Adapter 2 because it would find and match mapping b first. (No URLs would fall through if the order were reversed.)

Note that you can use only one wildcard (*) per URL.

To reach this screen for editing:
1. On the Main Menu under Application Integration Settings for My SP Configuration, click Adapters.
2. Click **Map URLs to Adapter Instances.**
   If this step does not appear, then you have created only one adapter instance (see “Configuring SP Adapters” on page 194).

**To create adapter mappings:**
1. Enter the URL and select an adapter from the drop-down menu.
2. Click **Add Mapping.**
3. Click **Save.**

**To edit adapter mappings:**
1. Click **Edit** next to the **Adapter Instance.** You can change the URL or select a different adapter from the drop-down menu.
2. Click **Update.**
3. Click **Save.**

**To delete adapter mappings:**
1. Click **Delete** next to the **Adapter Instance.**
2. Click **Save.**
   (Click **Cancel** to abort the deletion.)

**To change the order of adapter mappings:**
1. Click the up or down arrows at the left to rearrange the order.
2. Click **Save.**

### Configuring Default URLs

As an SP, you can supply a default URL that the end-user may see when SSO succeeds (that is, a session is created at your site) but the target resource is not available or not specified. Similarly, you can specify a default URL indicating a successful SLO to the end-user (if no other page is designated).

Your application or your partner’s application may supply these URLs at runtime (see “SP Endpoints” on page 338); but if none is provided, PingFederate will use the default values you enter on this screen.

**Tip:** If you leave the default URLs blank, PingFederate provides built-in landing pages for the user. These Web pages are among the templates you can modify with your own branding or other information (see “Customizing User-Facing Screens” on page 52).
Setting Target Validation

Several SP adapters can be configured to pass security tokens or other user credentials from PingFederate to the target resource via HTTP query parameters or POST transmittal. In both cases, these transport methods open the possibility that a third party (with specific knowledge of aspects of the IdP and/or SP network, as well as PingFederate endpoints and configuration) might be able to obtain and use valid security tokens to gain improper access to the target resource.

This potential security vulnerability would involve using well-formed SSO links to start an SSO request for a resource at the SP site. However, the target resource designated in the link would be intended to intercept the security token by redirection to a malicious Web site.

To prevent such an attack, PingFederate provides a means of validating SSO transactions to ensure that the designated target resource exists in a domain controlled by the SP.

Note: This configuration is optional, though recommended, if you have an SP adapter configured to use query parameters or POST transmittals. By default, however, most adapter configurations use cookies to pass session tokens. Such an adapter configuration does not pose any known security risks—target validation is not needed.
Chapter 6
Service Provider SSO Configuration

To enable target resource validation:
1. Click the first checkbox.
2. Indicate whether to require HTTPS for query and POST transmittals.

Note: This selection is recommended to ensure that target validation will always prevent message interception for this type of potential attack, under all conceivable permutations.

3. Enter the domain or IP address containing a target resource and click Add.
   Use the domain only, without qualifiers. For example:
   
   mycompany.com
   
   Using an initial wildcard and period for a domain name will cover multiple subdomains. For example:
   
   *.mycompany.com
   covers hr.mycompany.com or email.mycompany.com.

4. Repeat the previous step as needed.

Viewing Application Endpoints

Click Application Endpoints on the Main Menu to see a list of endpoints and descriptions applicable to your federation role (IdP or SP). These endpoints are built into PingFederate and cannot be changed.

Web-application developers at your site need to know the application endpoints to initiate transactions via PingFederate (see “SSO Integration Kits and Adapters” on page 4).

Note: For specific parameters required or allowed for Application Endpoints, see “SP Endpoints” on page 338.
This screen also shows a Maintenance Endpoint that you can use to verify that the PingFederate server is running (see “Maintenance Endpoint” on page 343).

**Federation Settings**

If your identity federation uses the SAML 2.0 XASP profile (see “Attribute Query and XASP” in the “Supported Standards” chapter of *Getting Started*), you may need to identify the IdP connection to which an attribute request applies. If so, click **Attribute Requester Mapping** under the Federation Settings section for the SP on the Main Menu.

Also under Federation Settings, you can view protocol endpoints that your federation partners need to know to access your services via PingFederate.

**Attribute Requester Mapping**

If you are using the XASP profile, the application(s) at your site must supply the Subject Distinguished Name (DN) to identify a user’s X.509 authentication certificate (see “Attribute Query and XASP” in the “Supported Standards” chapter of *Getting Started*). Optionally, an application may also supply an Issuer DN, which can be used to determine the correct IdP (Attribute Authority) to use for a set of users associated with an IdP.

**Note:** A `Format` query parameter must be set to a specified value for XASP (see “/sp/startAttributeQuery.ping” on page 342).

On the Attribute Requester Mapping screen, you can map X.509 identifying information to connections and specify a default connection. You reach this screen from the Main Menu under Federation Settings.

**Note:** The Attribute Requester Mapping link does not appear on the Main Menu unless you have enabled the SAML 2.0 protocol for the SP role (see “Choosing Roles and Protocols” on page 64). You must also select the associated XASP checkbox.
At runtime, PingFederate tries to match the certificate’s Issuer DN (if provided) against the list of Issuer DN(s), in the order shown on this screen, until a match is found. If no match is found, the server tries the Subject DN(s) in order. If no match is found, the Default connection is used.

For Issuer and Subject DNs, you can use a regular expression to match different DNs to the same connection. Only one expression may be used in any single entry. DN values must be entered in all lower-case characters.

**To map attribute requesters to connections:**
1. (Optional) Enter an Issuer DN when applicable, select a SAML 2.0 IdP Connection Name, and click Add.
   Repeat this step as needed for additional DNs.
2. Enter an Subject DN, select a SAML 2.0 IdP Connection Name, and click Add.
   Repeat this step as needed for additional DNs.
3. Select a Default IdP connection.

**To edit a mapping:**
1. Click Edit for the mapping in the Action column.
2. Make your changes and click Update in the Action column.
3. If you are editing an existing configuration, click Save to confirm the change.

**To reorder the mapping list:**
- Click the up or down arrow next to a DN.
To delete a mapping:
1. Click **Delete** for the mapping in the Action column.
2. If you are editing an existing configuration, click **Save** to confirm the deletion.

Viewing Protocol Endpoints

Click Protocol Endpoints under Federation Settings in the SP Configuration section of the Main Menu to see a list of SAML, WS-Federation, and/or WS-Trust STS endpoints—a pop-up window displays only those endpoints related to the federation protocols you have chosen (see “Choosing Roles and Protocols” on page 64). These endpoints are built into PingFederate and cannot be changed.

Your federation partners or STS clients need to know the applicable SP Services endpoints to communicate with your PingFederate server. Configured service endpoints for SAML connections are included in metadata export files (see “Exporting Metadata” on page 27).

The table below describes each endpoint:

<table>
<thead>
<tr>
<th>Service</th>
<th>URL and Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single Logout Service (SAML 2.0)</td>
<td>/sp/SLO.saml2&lt;br&gt;The URL that receives and processes logout requests and responses.</td>
</tr>
<tr>
<td>Assertion Consumer Service (SAML 2.0)</td>
<td>/sp/ACS.saml2&lt;br&gt;A SAML 2.0 implementation that receives and processes assertions from an IdP. The numbers reflect the index value PingFederate uses to handle each binding.</td>
</tr>
<tr>
<td>Artifact Resolution Service (SAML 2.0)</td>
<td>/sp/ARS.ssaml2&lt;br&gt;The SOAP endpoint that processes artifacts returned from a federation partner to retrieve the referenced XML message on the back channel.</td>
</tr>
<tr>
<td>Metadata Service</td>
<td>/&lt;br&gt;The default endpoint (empty path) from which partners can retrieve Auto-Connect metadata (see “Using Auto-Connect” on page 15).</td>
</tr>
<tr>
<td>Assertion Consumer Service (SAML 1.x)</td>
<td>/sp/acssaml1&lt;br&gt;A SAML 1.x implementation URL that receives and processes assertions from an IdP.</td>
</tr>
<tr>
<td>Single Sign-on Service (WS-Federation)</td>
<td>/sp/prp.wsf&lt;br&gt;The WS-Federation implementation URL that receives and processes security tokens and SLO messages.</td>
</tr>
</tbody>
</table>
Managing IdP Connections

As an SP, you manage connection settings to support the exchange of federation-protocol messages (SAML, WS-Federation, or WS-Trust) with an IdP or STS client application at your site.

**Note:** If you are configuring a new connection only for WS-Trust STS, follow the sections in this part of the manual up to and including “General Information” on page 212. Then turn to “WS-Trust STS Configuration” on page 269.

These settings include:

- User attributes you expect to receive in an SSO assertion (including STS SAML tokens).
- User attributes that may be requested using the Attribute Query profile (if that profile is used).
- The protocol and, for SAML, the profile you will use, including detailed security specifications (the use of digital signatures, signature verification, XML encryption, and SSL). For more information see the “Supported Standards” chapter in Getting Started.

To continue with the configuration, you and your connection partner must have decided this information in advance (see “Federation Planning Checklist” on page 19). Your federation partner must supply some connection settings and other information (see “Configuration Data Exchange” on page 21).

**Tip:** If you are configuring connections to more than one partner under SAML 2.0 specifications, or if you intend to add partners in the future, consider using Auto-Connect (see “Configuring IdP Auto-Connect” on page 265).

As an SP, you respond to user requests for SSO and SLO by creating or closing user sessions, respectively, in local applications. You integrate these applications with PingFederate by configuring them with SP adapter instances (see “Configuring SP Adapters” on page 194). In preparation for configuring a new SSO connection, you will need to know which adapter instance to use (see “Configuring Adapter Mapping and User Lookup” on page 222). (No adapters are required for a connection that uses only the Attribute Query profile—see “Configuring the Attribute Query Profile” on page 252.)

If you intend to pass attribute values to an adapter instance from a local data store, you must define the data store during this configuration, if you have not done so already (see “Managing Data Stores” on page 72).
Accessing Connections

You can create or modify connections directly via the Main Menu. Note that the menu displays the four most-recently modified connections. To view a list of all IdP connections, click the Manage All IdP link.

Using the Main Menu

From the Main Menu, you can configure a new connection, modify an existing connection, or view connections.

Tip: To copy or delete connections or to find connection drafts, click Manage All IdP (see “Using the Manage Connections Screen” on page 208).

Note that long connection names are truncated for this display. The full connection names are displayed on the Manage Connections screen (see “Using the Manage Connections Screen” on page 208).

To begin configuring a new connection:

1. Click Create New under IdP Connections on the Main Menu.

Tip: If you want to use a virtual ID for a second connection to the same partner, the fastest way is to click Manage All IdP and copy the first connection (see “Using the Manage Connections Screen” on page 208). For information about virtual IDs, see “Federation Server Identification” on page 20.

To modify a connection:

1. Click the connection name under IdP Connections on the Main Menu.

   Only the four most recently edited connections are displayed. To see all connections, including drafts, click Manage All IdP.

2. On the Activation & Summary screen, click the heading for the information you want to change.
3. Make your change and click **Save**.

**Note:** If **Save** is not available, it means your modification requires other changes or you are editing a screen that is part of a series of subtasks. Click **Next** and continue making indicated changes. The **Done** button indicates that further changes in the task are optional. When you have no further changes, click **Done** and then click **Save** on the task summary screen.

### Using the Manage Connections Screen

From the Manage Connections screen you can configure a new connection, modify or copy an existing connection or draft, or delete a connection (if it is not active). In addition, the screen provides export functions that allow you to save individual connection configurations or metadata about a connection that can be used to expedite your partner's corresponding configuration (see “Exporting Metadata” on page 27).

**Note:** The connection export function results in an XML file that you can modify and import into another PingFederate server acting in the same federation role (IdP or SP) at your site (see “Connection Management Service” on page 345). You can also automate this process (see “Automating Configuration Migration” on page 46).

Finally, from this screen you can also globally override transaction logging levels set for individual connections or restore connection-based logging (see “Runtime Transaction Logging” on page 26).

<table>
<thead>
<tr>
<th>Connection Name</th>
<th>Connection ID</th>
<th>Virtual ID</th>
<th>Protocol</th>
<th>Status</th>
<th>Logging Mode</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID Provider</td>
<td>id_provider</td>
<td>SAML2</td>
<td>ACTIVE</td>
<td>In Use / Copy / Export Connection / Export Metadata</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RealProperty</td>
<td>realproperty</td>
<td>SAML2</td>
<td>INACTIVE</td>
<td>Delete / Copy / Export Connection / Export Metadata</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**To access the Manage Connections screen:**

- Click **Manage All IdP** under IDP Connections on the Main Menu.

**To begin configuring a new connection:**

- Click **Create Connection** on the Manage Connections screen.

Managing IdP Connections

**Tip:** If you need to create a second connection to a partner using a Virtual ID, copy the existing connection and make necessary changes, including adding the Virtual ID on the General Info screen. For information about Virtual IDs, see “Federation Server Identification” on page 20.

To copy a connection:
1. Click **Copy** under Action for the connection you want to copy.
2. Enter new General Information for the connection (see “General Information” on page 212).
3. Make any further changes needed for the new connection.

To edit a connection or continue working on a draft:
- Click the **Connection Name** link.
  - For a draft, you will return to where you left off.

To export a connection:
1. Click **Export Connection** under Action for the connection.
2. Save the XML file on your file system.
  - You can change the name of the file, but keep the XML extension.

**Tip:** You can import the connection programmatically or manually into another instance of PingFederate acting in the same role (see “Connection Management Service” on page 345).

To export connection metadata:
1. Click **Export Metadata** under Action for the connection.
   - This action takes you to the Export Metadata screen flow, with the connection selection preset (see “Exporting Metadata” on page 27).
2. Complete the steps remaining in the Export Metadata screen flow (starting at Step 4 under “To export connection metadata:” on page 28).

To delete a connection:
1. Under Action, click **Delete** for the connection.
   - (To undo the deletion, click **Undelete**.)
   - **Note:** The Delete function is not available if the connection is active.
2. To confirm the deletion, click **Save**.

To sort the list of connections:
- Click the arrow next to any column heading to sort the list based on that column.

To filter the list by Protocol and/or Status:
- Select a filter criterion from either or both of the drop-down lists.

To override connection-based transaction logging:
1. Select **On** under Logging Mode Override.
2. Choose the logging mode you want to use for all connections.
To restore connection-based transaction logging:

- Select Off under Logging Mode Override.

Choosing a Connection Type

Indicate on the Connection Type screen whether the connection to this partner is for Browser SSO, WS-Trust STS, or both (see “Connection Types” on page 1).

**Note:** You can add STS support to any existing SSO connection, or vice versa, at any time.

Choosing Connection Options

On the Connection Options screen, you can choose to enable User Provisioning in conjunction with Browser SSO (see “Express Provisioning” on page 18).
Managing IdP Connections

For SAML 2.0, you also have the option of configuring the Attribute Query profile, with or without SSO (see “Attribute Query and XASP” in the “Supported Standards” chapter of Getting Started).

**Note:** This screen is presented only for browser-based SSO connections (see “Choosing a Connection Type” on page 210).

To create a connection for secure Internet SSO, ensure that Browser SSO is selected and click Next.

### Importing Metadata

If you are using one of the SAML protocols and have received a metadata file from your partner, click Browse on the Import Metadata screen, select the file, and click Next. For more information, see “Metadata” in the “Supported Standards” chapter of Getting Started.

**Note:** If the endpoints in the metadata file share the same base URL (protocol, hostname, and port), PingFederate uses this information to populate the Base URL field (see “General Information” on page 212). Consequently, individual endpoints on other screens do not include this information—only relative paths are shown.

**Note:** If you are importing a signed metadata file that does not include the certificate and public key, you will be asked to import the certificate needed to verify the XML signature (see the next section).

If you are not using a metadata file, click Next on the Import Metadata screen.

### Importing a Verification Certificate

The Import Certificate screen appears only if the metadata file you have chosen to import is signed and the certificate needed to verify the signature is not contained in the file.

- Click Browse to locate and open the signature verification certificate for this partner.
Chapter 6
Service Provider SSO Configuration

Viewing the Metadata Summary

The Metadata Summary screen provides security information about an imported metadata file, including whether the file was signed and, if so, the trust status of the certificate used to verify the signature.

General Information

On the General Info screen, you provide a required unique identifier and display name for a connection, as well as optional contact information. In addition, on this screen you can define a default error message that end users will see in the event that SSO fails, and you can set the level of transaction logging for this connection partner (see “Runtime Transaction Logging” on page 26).
## Field Descriptions

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Partner’s Entity ID/Issuer/Partner’s Realm (Connection ID)</td>
<td>(Required) The published, protocol-dependent, unique identifier of your partner. For a SAML 2.0 connection, this is your partner’s SAML Entity ID. For a SAML 1.x connection, this is the Issuer your partner advertises. For a WS-Federation connection, this is your partner’s Realm. This ID may have been obtained out-of-band or via a metadata file if you are using a SAML protocol (see “Exporting Metadata” on page 27). For STS-only connections, this ID can be any unique identifier.</td>
</tr>
<tr>
<td>Connection Name</td>
<td>(Required) A plain-language identifier for the connection—for example, a company or department name. This name is displayed in the connection list on the Main Menu.</td>
</tr>
<tr>
<td>Virtual Server ID</td>
<td>Enter a unique server ID in this field if you want to identify your server to this connection partner using an ID other than the one you specified under Server Settings (see “Specifying Federation Information” on page 67).</td>
</tr>
<tr>
<td>Base URL</td>
<td>The fully qualified hostname and port on which your partner’s federation deployment runs (e.g., <a href="https://www.pingidentity.com:9031">https://www.pingidentity.com:9031</a>). This entry is an optional convenience, allowing you to enter relative paths to specific endpoints, instead of full URLs, during the configuration process.</td>
</tr>
<tr>
<td>Company</td>
<td>The name of the partner company to which you are connecting.</td>
</tr>
<tr>
<td>Contact Name</td>
<td>The contact person at the partner company.</td>
</tr>
<tr>
<td>Contact Number</td>
<td>The phone number of the contact person at the partner company.</td>
</tr>
<tr>
<td>Contact Email</td>
<td>The email address for the contact person at the partner company.</td>
</tr>
<tr>
<td>Error Message</td>
<td>If an error occurs on this server, the end user’s browser may be redirected (in a default situation) to an error page hosted within . The text entered here is shown on that page. (This entry is not displayed for STS-only connections.)</td>
</tr>
<tr>
<td>Logging Mode</td>
<td>The level of transaction logging applicable for this connection (see “Runtime Transaction Logging” on page 26). Note that you can override connection logging mode settings globally from the connections list (see “Using the Manage Connections Screen” on page 208).</td>
</tr>
</tbody>
</table>
Chapter 6
Service Provider SSO Configuration

For a new connection:
- Fill in the information needed and click Next.
  Connection ID and Connection Name are required (see “Field Descriptions” above).
  Note that the Virtual ID identifies your own federation deployment for this connection only and overrides the ID you specified under Server Settings (see “Federation Server Identification” on page 20).

For an existing connection:
- If you are editing existing information, modify the fields as needed and click Save.

Configuring Browser-Based SSO

Browser-based SSO (also, Browser SSO) is another term for secure Internet SSO, which relies on a user’s Web browser and HTTP to broker XML identity-federation messaging between an IdP and an SP (in contrast to WS-Trust STS messaging, which is typically application-driven across the back channel and does not require browser mediation).

To continue, click Configure Browser SSO.

Configuration Steps

Many steps involved in setting up a federation connection are protocol-independent; that is, they are required steps for all connections, regardless of the associated standards (see the “Supported Standards” chapter in Getting Started). Also, for any given connection, some configuration steps are required under the applicable protocol, while others are optional. Still others are required only based on certain selections. The PingFederate administrative console determines the required and optional steps based on the protocol and dynamically presents additional requirements or options based on selections.
Managing IdP Connections

The following sections provides sequential information about every step you might encounter while configuring browser-based SSO, regardless of the protocol you are using for a particular connection.

**Note:** The configuration screens represented in this chapter show “SAML 2.0” in their left corners, unless they are exclusive to WS-Federation or SAML 1.x setup requirements. When the SAML 2.0 screens are also applicable to SAML 1.x and/or WS-Federation connections, the SAML 2.0 representations and discussions also apply to the other protocols, unless otherwise indicated.

After configuring SSO settings, you will normally need to configure authentication credentials, the range of which depends on your SSO selections (see “Configuring Credentials” on page 254). Also, other configuration tasks may remain to be configured for new or modified connections, depending on selected connection options (see “Choosing Connection Options” on page 210).

**Important:** For new connections you must completely configure these SSO settings and subsequent tasks before you can save the connection on the Activation & Summary screen. Until then, the configuration is temporary and can be lost; the console times out after several minutes of inactivity. At any time, however, you can click **Save Draft**, which is available on most screens after you enter General Information (see “Console Buttons” in the “Console Navigation” chapter of *Getting Started*).

Use the lists and links (or page references) below to find specific information about steps that may apply to your SSO connection requirements:

**SAML 2.0 SSO Configuration Steps**
- “Choosing SAML Profiles” on page 216
- “User-Session Creation” on page 218
  - “Selecting an Identity-Mapping Method” on page 219
  - “Creating an Attribute Contract” on page 220
  - “Configuring Adapter Mapping and User Lookup” on page 222
- “Configuring Protocol Settings” on page 234
  - “Specifying SSO Service URLs (SAML)” on page 235
  - “Specifying SLO Service URLs (SAML 2.0)” on page 237
  - “Choosing Allowable SAML Bindings (SAML)” on page 238
  - “Setting an Artifact Lifetime (SAML 2.0)” on page 239
  - “Specifying Artifact Resolver Locations” on page 240
  - “Configuring Signature Policy” on page 241
  - “Configuring XML Encryption Policy (SAML 2.0)” on page 242

**WS-Federation SSO Configuration Steps**
- “User-Session Creation” on page 218
  - “Selecting an Identity-Mapping Method” on page 219
  - “Creating an Attribute Contract” on page 220
  - “Configuring Adapter Mapping and User Lookup” on page 222
- “Configuring Protocol Settings” on page 234
  - “Specifying a Service URL (WS-Federation)” on page 236
Chapter 6
Service Provider SSO Configuration

SAML 1.x SSO Configuration Steps

- “Choosing SAML Profiles” on page 216
- “User-Session Creation” on page 218
  - “Selecting an Identity-Mapping Method” on page 219
  - “Creating an Attribute Contract” on page 220
  - “Configuring Adapter Mapping and User Lookup” on page 222
- “Configuring Protocol Settings” on page 234
  - “Specifying SSO Service URLs (SAML)” on page 235
  - “Choosing Allowable SAML Bindings (SAML)” on page 238
  - “Specifying Artifact Resolver Locations” on page 240
  - “Configuring Signature Policy” on page 241

Choosing SAML Profiles

A SAML profile is the message-interchange scenario that you and your federation partner have agreed to use (see “Federation Planning Checklist” on page 19). For SAML 2.0, PingFederate supports all IdP- and SP-initiated SSO and SLO profiles.

The SAML 1.x implementation supports standard IdP-initiated SSO as well as nonstandard SP-initiated SSO.

For information on typical SSO/SLO profile configurations, including illustrations, see the “Profiles” sections in the “Supported Standards” chapter in Getting Started.

You can configure profiles individually or all together. PingFederate presents the correct configuration steps to fit your choices. Steps that apply to one SSO or SLO profile often apply to others and are reused automatically across profiles.

Note: For SAML 1.x, IdP-initiated SSO is assumed and the specifications do not support SLO; the only choice on this screen is SP-initiated SSO (see “SAML 1.x Profiles” in the “Supported Standards” chapter of Getting Started).

For WS-Federation, the SAML Profiles screen is not presented.
Managing IdP Connections

To reach this screen:
1. Click a connection name on the Main Menu.
   Click Manage All IdP, if needed, to see a full list of connections.
2. Click Browser SSO under the IdP Connection tab.
3. Click SAML Profiles on the Summary screen.

To configure profiles:
1. Select the profile(s) applicable to this connection and click Next.
   For SAML 2.0 connections, you must select an SSO profile before you can enable SLO.
2. Continue through the remaining connection-configuration tasks.

Configuring IdP-Initiated SSO

When PingFederate is operating as an SP, the IdP-initiated SSO profile configuration defines the message-transport mechanisms (bindings) your enterprise has agreed to allow for receiving SAML assertions, plus any digital signature verification requirements for inbound assertions (see “Certificates, SSL, and XML Encryption” on page 10).

For illustrations of typical profile and binding scenarios, see the “Profiles” sections in the “Supported Standards” chapter in Getting Started.

For this configuration you need to know:
• The transport binding(s) to which you and your partner have agreed
• The certificate to be used for verifying incoming digital signatures from your IdP (optional for the artifact binding)

When Artifact is an allowable inbound SAML binding, you also need to know the endpoint(s) to your partner’s Artifact Resolution Service(s) and the SOAP client authentication mechanism to use: either HTTP Basic, SSL client certificates, a digital signature, or a combination of these mechanisms.

Configuring SP-Initiated SSO

The SP-initiated SSO profile configuration defines the message-transport mechanisms (bindings) and security requirements for sending authentication requests and receiving assertions when your site initiates SSO transactions.

For SAML 1.0, the SP-initiated SSO profile is also known as the “destination-first” profile, which was added as a supported “non-normative” use case after the release of the SAML 2.0 specifications.

For illustrations of typical profile and binding scenarios, see the “Profiles” sections in the “Supported Standards” chapter in Getting Started.

For this configuration you will need to know:
• The endpoint URL(s) for your IdP’s Single Sign-on Service(s)
• The transport bindings to which you and your partner have agreed (inbound and outbound)
• The certificates you will use to sign outbound authentication requests and to verify incoming digital signatures from your IdP

When Artifact is an allowable inbound SAML binding, you also need to know the endpoint(s) to your partner’s Artifact Resolution Service(s) and the SOAP client authentication mechanism to use: either HTTP Basic, SSL client certificates, a digital signature, or a combination of these mechanisms.
Configuring IdP-Initiated SLO

The SAML 2.0 IdP-initiated SLO profile configuration defines the message-transport mechanisms (bindings) and security requirements that you and your partner have agreed upon for exchanging SLO requests and responses.

\[\text{Note: SLO is not supported by the SAML 1.x specifications.}\]

For more information about SLO, see “Single Logout” in the “Supported Standards” chapter of Getting Started.

For this configuration you need to know:

- The transport bindings that you and your partner have agreed upon to send SLO requests and receive responses
- The certificates to be used for signing outgoing messages and for verifying incoming digital signatures from your IdP (optional for the artifact binding)
- The URL(s) of your IdP’s Single Logout Service(s)
- The URL of your IdP’s Artifact Resolution Service(s) (to resolve artifacts from the IdP) and SOAP client authentication requirements

Configuring SP-Initiated SLO

The SAML 2.0 SP-initiated profile configuration for SLO defines the message-transport mechanisms (bindings) and security requirements that you and your partner have agreed upon for exchanging SAML requests and responses.

\[\text{Note: SLO is not supported by the SAML 1.x specifications.}\]

For more information about SLO, see “Single Logout” in the “Supported Standards” chapter of Getting Started.

For this configuration you need to know:

- The transport bindings that you and your partner have agreed upon to send SLO requests and receive responses
- The certificates to be used for signing outgoing messages and for verifying incoming digital signatures from your IdP (optional for the artifact binding)
- The URL(s) of your IdP’s Single Logout Service(s)
- The URL of your IdP’s Artifact Resolution Service(s) (to resolve artifacts from the IdP) and SOAP client authentication requirements

User-Session Creation

As an SP, you must specify how you will use information sent from the IdP in SSO assertions to create user sessions for enabling access to protected resources at your site. This configuration includes:

- Choosing an identity-mapping method (see “Selecting an Identity-Mapping Method” next).
- Defining the attribute contract you will use with this partner, if any (see “Creating an Attribute Contract” on page 220).
- Configuring instances of one or more adapters (see “Configuring Adapter Mapping and User Lookup” on page 222) and specifying how they are used to fulfill the adapter contract.
To continue, click **Configure User-Session Creation**.

### Selecting an Identity-Mapping Method

PingFederate allows an SP to use either **account linking** or **account mapping** to associate remote users with local accounts for SSO between business partners (see “Identity Mapping” on page 5). At the Identity Mapping step, you choose which method to use with a particular IdP connection. You and your partner may want to decide in advance which option to use (see “Federation Planning Checklist” on page 19).

- **Account Linking**: The IdP is sending a unique name identifier (possibly opaque). An opaque identifier preserves user privacy in that it cannot be traced back to a user's identity at the IdP. The name identifier is used by this SP to create a persistent association between the user and a specific local account.
- **Account Mapping**: The SP is sending a set of attributes that may be used to dynamically map the user to a specific local account.

If your site is using account linking, then establishing an **attribute contract** is not required. Depending on your partner agreement, however, you may choose to supplement the account link with an attribute contract. In this configuration the account link is used to determine the user's identity, while the additional attributes
might be used for authorization decisions, customized Web pages, and so on, at your site (see “About Attributes” on page 6).

---

**Important:** If you have previously set up a configuration to use an attribute contract and want to change the configuration to use account linking without additional attributes, then the existing attribute contract will be discarded.

---

Account linking can be used with either a clear, standard name identifier or an opaque pseudonym.

► If you want to dynamically associate remote users with local accounts using a known attribute to identify a user—for example, a username or email address—then select **Account Mapping**.

Account mapping uses the value passed in the SAML assertion's `SAML_SUBJECT` and associated user attributes to create an association between a remote user and a local account.

---

**Tip:** if you are using PingFederate’s Express Provisioning, choose Account Mapping (see “User Provisioning” on page 244).

► If you want to create a long-term association between a remote user and a local account, then select **Account Linking** on the Identity Mapping screen.

To set up an attribute contract to use in conjunction with account linking, click the checkbox next to “The assertion includes attributes . . .” after selecting **Account Linking**.

---

**To reach this screen for editing:**

1. Click the connection name on the Main Menu.
   - Click **Manage All IdP**, if needed, to see a full list of connections.
2. Click **Browser SSO** under the IdP Connection tab.
3. Click **Configure Browser SSO**.
4. Click **User-Session Creation** under the Browser SSO tab.
5. Click **Configure User-Session Creation**
6. Click **Identity Mapping** on the Summary screen.

**Creating an Attribute Contract**

An attribute contract is the set of user attributes that you and your partner have agreed will be sent in SAML assertions for this connection (see “Attribute Contracts” on page 7). You identify these attributes on this screen.

`SAML_SUBJECT` is always sent in a SAML assertion and contains the name identifier of the user requesting SSO. When you select **account mapping** as the identity mapping technique, the `SAML_SUBJECT` is available to help map the incoming user to a local ID on your system (see “Selecting an Identity-Mapping Method” on page 219).

For **account linking**, the `SAML_SUBJECT` contains an identifier that the SP server uses to make a permanent association between the remote user and a local account. The `SAML_SUBJECT` itself is not available to the SP application and thus does not appear in the Attribute Contract on this screen.
Optionally, you can mask the values of attributes (other than SAML_SUBJECT) in the log files that PingFederate writes when it receives assertions (see “Attribute Masking” on page 9).

To reach this screen for editing:
1. Click the connection name on the Main Menu.
   Click Manage All IdP, if needed, to see a full list of connections.
2. Click Browser SSO under the IdP Connection tab.
3. Click Configure Browser SSO.
4. Click User-Session Creation under the Browser SSO tab.
5. Click Configure User-Session Creation
6. Click Attribute Contract on the Summary screen.

   If this step is not in the list, then you have chosen to use account linking and specified that the IdP is not including additional attributes in the assertion (see “Selecting an Identity-Mapping Method” on page 219).

To add an attribute:
1. Enter the attribute name in the text box.
   Attribute names are case-sensitive and must correspond to the names expected by your partner.
2. (Optional) Select the checkbox under Mask Values in Log.
3. Click Add.

To modify an attribute name or masking selection:
1. Click Edit under Action for the attribute.
2. Make the change and click Update.

   Note: If you change your mind, ensure that you click the Cancel link in the Actions column, not the Cancel button, which discards any other changes you might have made in the configuration steps.

To delete an attribute:
1. Click Delete under Action for the attribute.
Configuring Adapter Mapping and User Lookup

Remote users arriving at your site via an SSO request do so in order to use specific applications or gain access to protected resources. Based on the nature of the business relationship and the agreement with your partner, you may be expected to provide access to these applications. Therefore, integration between your federation SP server and local applications is important.

The PingFederate server for an SP uses integration adapters to identify the user to your applications based on attributes sent in an assertion. The server uses this information to create a local session that enables access to user-requested resources (the “target”) at your site. (See “SSO Integration Kits and Adapters” on page 4.)

Each adapter instance requires a set of attributes into which you map values found in the assertion. You can map additional attributes, as needed, from local data stores, or you can use static or variable text values. An adapter instance will fail to create a local session for the incoming user if it is unable to find values for each of its required attributes.

You must associate at least one adapter instance with an IdP connection. If you have multiple integration requirements—for example, if you are using more than one IdM system or an application not covered by a centralized system—then you should map multiple adapter instances.

**Note:** If you configure only one adapter instance for a connection, the server will use that instance at runtime without checking for any associated URLs (see “Mapping URLs to Adapter Instances” on page 198).

---

**To reach this screen for editing:**
1. Click the connection name on the Main Menu.
   - Click **Manage All IdP**, if needed, to see a full list of connections.
2. Click **Browser SSO** under the IdP Connection tab.
3. Click **Configure Browser SSO**.
4. Click **User-Session Creation** under the Browser SSO tab.
5. Click **Configure User-Session Creation**
6. Click **Adapter Mapping & User Lookup** on the Summary screen.
To begin mapping an adapter:

- Click Map New Adapter Instance and follow the configuration screens (see the following sections for more information).

To begin modifying an existing adapter mapping:

- Click the Adapter Instance Name and navigate through the steps to the information you want to change.

Selecting an Adapter Instance

An SP adapter instance is available for use within an IdP connection only after it has been deployed in PingFederate.

To select an adapter instance:

- Choose an adapter instance from the drop-down menu and click Next to continue.

  If the adapter instance you need is not available, click Manage Adapter Instances to define one or more adapter instances you need for this connection.

  Note that an adapter instance can be mapped only once per connection.

  **Tip:** Adapter contracts for some adapters can be customized for individual connection requirements (see “Configuring SP Adapters” on page 194).

Selecting an Adapter Data Store

To populate the attributes required by the adapter (the adapter contract), you can use values supplied by SAML assertions from the IdP exclusively, or in addition to values retrieved from local user data stores (see “Managing Data Stores” on page 72).
If you choose to look up additional values, click the applicable button and then **Next**. This selection allows you to identify data sources and specify lookup queries in subsequent screens (see “Data Store Setup” next).

Or:

If you choose not to look up additional values, click the applicable button (if it is not already selected) and then **Next**. This selection takes you directly to a screen where you can map attribute values from the assertion (see “Configuring Adapter Contract Fulfillment” on page 233).

**Tip:** To determine whether you need to look up additional values, compare your attribute contract against your adapter contract (see “Creating an Attribute Contract” on page 220 and “Selecting an Adapter Instance” on page 223). If the adapter requires more information, determine whether your local data stores can supply it. (You can also choose to use text constants or expressions for certain information—see “Configuring Adapter Contract Fulfillment” on page 233.)

---

### Data Store Setup

For data-store setup information, refer to the sections indicated in the following steps.

**Note:** As you make selections on configuration screens, ensure that you allow enough time for PingFederate to access your data store and populate drop-down lists.

1. See “Selecting a Data Store” on page 225.
2. See the following sections in this manual, depending on the type of data store:

<table>
<thead>
<tr>
<th>Data Store Type</th>
<th>Related Manual Sections</th>
</tr>
</thead>
</table>
| JDBC            | • “Selecting a Database Table and Columns” on page 226  
                  • “Configuring a Database Filter (WHERE Clause)” on page 227 |
Managing IdP Connections

3. See “Configuring Adapter Contract Fulfillment” on page 233

Selecting a Data Store

This screen allows you to choose a data store from a previously configured list (see “Managing Data Stores” on page 72). Attribute values extracted from this data store will be used in combination with the values from the attribute contract to fulfill the adapter contract for this adapter instance (see “Configuring Adapter Mapping and User Lookup” on page 222).

<table>
<thead>
<tr>
<th>Data Store Type</th>
<th>Related Manual Sections</th>
</tr>
</thead>
</table>
| LDAP            | • “Configuring an LDAP Directory Search” on page 229  
|                 | • “Configuring an LDAP Filter” on page 231            |
| Custom          | • “Configuring Custom Source Filters” on page 232     
|                 | • “Selecting Custom Source Fields” on page 232        |

3. See “Configuring Adapter Contract Fulfillment” on page 233

Selecting a Data Store

This screen allows you to choose a data store from a previously configured list (see “Managing Data Stores” on page 72). Attribute values extracted from this data store will be used in combination with the values from the attribute contract to fulfill the adapter contract for this adapter instance (see “Configuring Adapter Mapping and User Lookup” on page 222).

<table>
<thead>
<tr>
<th>Data Store Type</th>
<th>Related Manual Sections</th>
</tr>
</thead>
</table>
| LDAP            | • “Configuring an LDAP Directory Search” on page 229  
|                 | • “Configuring an LDAP Filter” on page 231            |
| Custom          | • “Configuring Custom Source Filters” on page 232     
|                 | • “Selecting Custom Source Fields” on page 232        |

3. See “Configuring Adapter Contract Fulfillment” on page 233

Selecting a Data Store

This screen allows you to choose a data store from a previously configured list (see “Managing Data Stores” on page 72). Attribute values extracted from this data store will be used in combination with the values from the attribute contract to fulfill the adapter contract for this adapter instance (see “Configuring Adapter Mapping and User Lookup” on page 222).

<table>
<thead>
<tr>
<th>Data Store Type</th>
<th>Related Manual Sections</th>
</tr>
</thead>
</table>
| LDAP            | • “Configuring an LDAP Directory Search” on page 229  
|                 | • “Configuring an LDAP Filter” on page 231            |
| Custom          | • “Configuring Custom Source Filters” on page 232     
|                 | • “Selecting Custom Source Fields” on page 232        |

3. See “Configuring Adapter Contract Fulfillment” on page 233

Selecting a Data Store

This screen allows you to choose a data store from a previously configured list (see “Managing Data Stores” on page 72). Attribute values extracted from this data store will be used in combination with the values from the attribute contract to fulfill the adapter contract for this adapter instance (see “Configuring Adapter Mapping and User Lookup” on page 222).

<table>
<thead>
<tr>
<th>Data Store Type</th>
<th>Related Manual Sections</th>
</tr>
</thead>
</table>
| LDAP            | • “Configuring an LDAP Directory Search” on page 229  
|                 | • “Configuring an LDAP Filter” on page 231            |
| Custom          | • “Configuring Custom Source Filters” on page 232     
|                 | • “Selecting Custom Source Fields” on page 232        |

3. See “Configuring Adapter Contract Fulfillment” on page 233

Selecting a Data Store

This screen allows you to choose a data store from a previously configured list (see “Managing Data Stores” on page 72). Attribute values extracted from this data store will be used in combination with the values from the attribute contract to fulfill the adapter contract for this adapter instance (see “Configuring Adapter Mapping and User Lookup” on page 222).

<table>
<thead>
<tr>
<th>Data Store Type</th>
<th>Related Manual Sections</th>
</tr>
</thead>
</table>
| LDAP            | • “Configuring an LDAP Directory Search” on page 229  
|                 | • “Configuring an LDAP Filter” on page 231            |
| Custom          | • “Configuring Custom Source Filters” on page 232     
|                 | • “Selecting Custom Source Fields” on page 232        |

3. See “Configuring Adapter Contract Fulfillment” on page 233

Selecting a Data Store

This screen allows you to choose a data store from a previously configured list (see “Managing Data Stores” on page 72). Attribute values extracted from this data store will be used in combination with the values from the attribute contract to fulfill the adapter contract for this adapter instance (see “Configuring Adapter Mapping and User Lookup” on page 222).

<table>
<thead>
<tr>
<th>Data Store Type</th>
<th>Related Manual Sections</th>
</tr>
</thead>
</table>
| LDAP            | • “Configuring an LDAP Directory Search” on page 229  
|                 | • “Configuring an LDAP Filter” on page 231            |
| Custom          | • “Configuring Custom Source Filters” on page 232     
|                 | • “Selecting Custom Source Fields” on page 232        |

3. See “Configuring Adapter Contract Fulfillment” on page 233

Selecting a Data Store

This screen allows you to choose a data store from a previously configured list (see “Managing Data Stores” on page 72). Attribute values extracted from this data store will be used in combination with the values from the attribute contract to fulfill the adapter contract for this adapter instance (see “Configuring Adapter Mapping and User Lookup” on page 222).

<table>
<thead>
<tr>
<th>Data Store Type</th>
<th>Related Manual Sections</th>
</tr>
</thead>
</table>
| LDAP            | • “Configuring an LDAP Directory Search” on page 229  
|                 | • “Configuring an LDAP Filter” on page 231            |
| Custom          | • “Configuring Custom Source Filters” on page 232     
|                 | • “Selecting Custom Source Fields” on page 232        |

3. See “Configuring Adapter Contract Fulfillment” on page 233
Selecting a Database Table and Columns

When you choose to use a database source for attributes, you follow this path through the configuration steps.

On this screen you specify the database column locations that will be retrieved after a lookup query locates the user record. You will specify the user lookup query next (see “Configuring a Database Filter (WHERE Clause)” on page 227).

Field Descriptions

Field Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schema</td>
<td>Lists the table structure that stores information within a database. Some databases, such as Oracle, require selection of a specific schema for a JDBC query. Other databases, such as MySQL, do not require selection of a schema.</td>
</tr>
<tr>
<td>Table</td>
<td>The name of the table contained in the database. The name is used to construct the SQL query to retrieve data from the data store.</td>
</tr>
<tr>
<td>Add Attribute</td>
<td>Adds the column to be executed in the SQL query to the data store to retrieve the attribute value.</td>
</tr>
<tr>
<td>(Drop-down menu)</td>
<td>The available columns of attribute names for the selected table are displayed. Select the columns that are associated with the desired attributes you would like to return from the JDBC query.</td>
</tr>
</tbody>
</table>

To reach this screen for editing:

1. Click the connection name on the Main Menu.
   Click Manage All IdP, if needed, to see a full list of connections.
2. Click Browser SSO under the IdP Connection tab.
3. Click Configure Browser SSO.
4. Click User-Session Creation under the Browser SSO tab.
5. Click **Configure User-Session Creation**

6. Click **Adapter Mapping & User Lookup** under the User-Session Creation tab.

7. Click the Adapter Instance Name.

8. Click **Database Tables and Columns** on the Summary screen.

   If this step is not shown, this connection is not yet configured to use a database to look up attributes. For information about changing this configuration, see “Selecting a Data Store” on page 225.

**To select a database table and columns for queries:**

1. Choose a Schema file (when applicable) from the drop-down list.
2. Choose a Table from the drop-down list.
3. Choose a name under Columns to Return from Select and click Add Column.

   Repeat this step for other columns as needed.

---

**Note:** You do not need to add a column here for it to be used as part of a search key (see “Configuring a Database Filter (WHERE Clause)” next). Add only attributes from which you need actual values to pass to the adapter.

---

**Tip:** To determine which attributes to look up during a query, click **Adapter Contract to Fulfill** to see what information must be collected (see “Selecting an Adapter Instance” on page 223). Then determine what information is coming in from the assertion (see “Creating an Attribute Contract” on page 220). Information not contained in the Attribute Contract may be pulled from the data store look-up query.

---

**Configuring a Database Filter (WHERE Clause)**

On this screen you begin to specify exactly where additional data can be found to complete the attribute contract when you receive an assertion from this IdP (see “Creating an Attribute Contract” on page 127).
The JDBC WHERE clause queries your data store to locate a user record. Once the record is located, the configured SELECT statements retrieve the attribute values.

The clause is in the form:
WHERE column1=value1 [AND column2=value2] [OR ...]

The left side of the first variable pair uses a column name in the database table you selected (see “Selecting a Database Table and Columns” on page 226).

The right side generally uses values passed in from the assertion. Possible variables for these, including the correct syntax, are listed under Assertion Values.

You can also apply additional search criteria from your own database, using any other columns from the targeted table.

**Tip:** Click “View List of Columns . . .” to see a list from which to copy and paste.

For general information about WHERE clauses, consult your DBMS documentation.

**Example:**

```plaintext
userid='$\{username\}'
```

In this example $\{username\}$ is the name of a column in the JDBC data store. On the right side, '$\{username\}' returns the value of the username from the assertion.

**Important:** You must use the $()$ syntax to retrieve the value of the enclosed variable and use single quotation marks around the $()$ characters.

To reach this screen for editing:

1. Click the connection name on the Main Menu.
   Click Manage All IdP, if needed, to see a full list of connections.
2. Click Browser SSO under the IdP Connection tab.
3. Click **Configure Browser SSO**.
4. Click **User-Session Creation** under the Browser SSO tab.
5. Click **Configure User-Session Creation**
6. Click **Adapter Mapping & User Lookup** under the User-Session Creation tab.
7. Click the Adapter Instance Name.
8. Click **Database Filter** on the Summary screen.

If this step is not shown, this connection is not yet configured to use a database to look up attributes. For information about changing this configuration, see “Selecting a Data Store” on page 225.

**To construct the WHERE clause:**

1. Enter the statement in the space provided, following the guidelines and example above.
   
   The initial WHERE is optional.
2. Ensure the syntax and variable names are correct.
   
   When you click Next, you will map attribute values returned from the database into the attribute contract (see “Selecting an Adapter Data Store” on page 223).

**Configuring an LDAP Directory Search**

When you choose to use an LDAP source for attributes, you follow this path through the configuration steps.

On this screen you begin to specify exactly where additional data can be found to supply to the SP adapter in order to access a resource on your system (see “SSO Integration Kits and Adapters” on page 4).
Field Descriptions

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base DN</td>
<td>The base distinguished name of the tree structure in which the search begins. This field is optional if records are located at the LDAP root.</td>
</tr>
<tr>
<td>Search Scope</td>
<td>Determines the node depth of the query. Select Subtree, One level or Object.</td>
</tr>
<tr>
<td>Root Object Class</td>
<td>Specifies the object type within the LDAP hierarchy from which attributes will be returned.</td>
</tr>
<tr>
<td>Attribute</td>
<td>The available attribute names for the selected directory structure. Select the names associated with the attributes that you would like to return from the query.</td>
</tr>
</tbody>
</table>

To reach this screen for editing:
1. Click the connection name on the Main Menu.
   - Click Manage All IdP, if needed, to see a full list of connections.
2. Click Browser SSO under the IdP Connection tab.
3. Click Configure Browser SSO.
4. Click User-Session Creation under the Browser SSO tab.
5. Click Configure User-Session Creation
6. Click Adapter Mapping & User Lookup under the User-Session Creation tab.
7. Click the Adapter Instance Name.
   - If this step is not shown, this connection is not yet configured to use LDAP to look up attributes (see “Selecting a Data Store” on page 225).

To select LDAP attributes:
1. (Optional) Enter a Base DN.
2. Select a Search Scope.
3. Select a Root Object Class.
4. Under Attributes to return from search, choose an attribute and click Add Attribute.
   - Note that the attribute Subject DN is always returned by default.
5. Repeat the last step for other attributes as needed.
6. (Optional) Change the Search Scope or the Root Object Class if you want attributes from other locations.

**Note:** You do not need to add an attribute here for it to be used in a search filter (see “Configuring an LDAP Filter”). Add only attributes from which you need values to map to the adapter.
Configuring an LDAP Filter

The LDAP filter queries the data store to retrieve a user record associated with a particular value (or values) from the assertion. The filter is in the form:

\[(attribute=${value})\]

The left-side variable is an attribute from the data store (see “Configuring an LDAP Directory Search” on page 229).

The right side generally uses values passed in from the assertion.

You can also apply additional search criteria from your data store, using any other attributes from the targeted object classes.

**Tip:** Click “View List of Available LDAP Attributes” for a list from which you can copy and paste.

For general information about search filters, consult your LDAP documentation.

**Example:**

\[(UNAME=${username})\]

In this example UNAME is the name of an attribute in the LDAP data store. On the right side, \(\{\text{username}\}\) returns the value of username in the assertion.

**Important:** You must use the \(\{\}\) syntax to retrieve the value of the enclosed variable.
Field Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filter</td>
<td>A filter narrows a search to locate requested data by either including or excluding specific records. An LDAP filter includes the attributes in the search and the value or range of values that the search is attempting to match. Searches are conducted by using at least three components: 1) at least one attribute (attribute data type) to search on, 2) a search filter operator that will determine what to match, and 3) the value of the attribute being sought. Searches must have at least one of each of these three components.</td>
</tr>
</tbody>
</table>

To reach this screen for editing:

1. Click the connection name on the Main Menu.
   - Click Manage All IdP, if needed, to see a full list of connections.
2. Click Browser SSO under the IdP Connection tab.
3. Click Configure Browser SSO.
4. Click User-Session Creation under the Browser SSO tab.
5. Click Configure User-Session Creation.
6. Click Adapter Mapping & User Lookup under the User-Session Creation tab.
7. Click the Adapter Instance Name.
8. Click LDAP Filter on the Summary screen.
   - If this step is not shown, this connection is not configured to use LDAP to look up attributes (see “Selecting a Data Store” on page 225).

To construct the LDAP filter:

1. Enter the statement in the space provided, following the guidelines and example above.

   Note: If you used an anonymous binding to create this LDAP connection, your access might be restricted (see “Configuring an LDAP Connection” on page 77).

2. Ensure the syntax and variable names are correct.
3. Click Next.

Configuring Custom Source Filters

When you choose to use a custom source for attributes, you follow this path through the configuration steps.

On this screen you specify a filter, or lookup query, for your custom data source. This screen display and the syntax of the filter depends on your developer's implementation of the custom source SDK.

Selecting Custom Source Fields

On the Configure Custom Source Fields screen, you can choose from among the fields shown to map to the adapter contract. These choices are supplied by the driver.
implementation. Select only those needed to fulfill the attribute contract for this partner connection.

**Configuring Adapter Contract Fulfillment**

The last step in configuring an adapter is to map each attribute required for the adapter contract to a value (see “Selecting an Adapter Data Store” on page 223). These are the values that will be used to create a local session. An SSO operation fails if the SP is unable to fulfill the mapping requirements defined here.

**Map attributes from one of these Sources:**

- **Account Link**
  
  This source appears only if you have elected to use account linking (see “Selecting an Identity-Mapping Method” on page 219). When you make this selection, the associated Value drop-down list is populated with Local User ID. Normally, you would map this identifier to target an adapter attribute that represents the local user ID.

- **Assertion**
  
  Values are contained in the assertions from this IdP. When you make this selection, the associated Value drop-down list is populated by the attribute contract (see “Creating an Attribute Contract” on page 220).

- **JDBC/LDAP/Custom**
  
  Values are returned from your query. When you make this selection, the Value list is populated by the database columns or LDAP or custom attributes you identified for this data store (see “Selecting a Database Table and Columns” on page 226, “Configuring an LDAP Directory Search” on page 229, or “Selecting Custom Source Fields” on page 232).

- **Expression (when enabled)**
  
  This option provides more complex mapping capabilities—for example, transforming incoming values into different formats (see “Using Attribute Mapping Expressions” on page 353). All of the variables available for text entries (see below) are also available for expressions.

- **Text**
  
  The value is what you enter. This can be text only, or you can mix text with references to any of the values from the assertion, using the \${attribute} syntax.

  You can also enter values from your data store, when applicable, using this syntax:

  \${ds.attribute}

  where attribute is any of the data store attributes you have selected.

  **Tip:** Two other variables are also available: \${SAML_SUBJECT} and \${TargetResource}. SAML_SUBJECT is the initiating user (or other entity). TargetResource is a reference to the protected application or other resource for which the user requested SSO access; this variable is available only if specified as a query parameter for the relevant PingFederate endpoint (either as TargetResource for SAML 2.0 or TARGET for SAML 1.x—see “Application Endpoints” on page 335).

  There are a variety of reasons why you might hard code a text value. For example, if your Web application provides a consumer service, you might want to supply a particular promotion code for this partner.
Chapter 6
Service Provider SSO Configuration

To reach this screen for editing:
1. Click the connection name on the Main Menu.
   - Click Manage All IdP, if needed, to see a full list of connections.
2. Click Browser SSO under the IdP Connection tab.
3. Click Configure Browser SSO.
4. Click User-Session Creation under the Browser SSO tab.
5. Click Configure User-Session Creation
6. Click Adapter Mapping & User Lookup under the User-Session Creation tab.
7. Click the Adapter Instance Name.
8. Click Adapter Contract Fulfillment on the Summary screen.

To map attributes:
1. Choose a Source for each Target attribute (see descriptions of each Source type above).
2. Choose (or enter) a Value for each Attribute.
   - All values must be mapped.
3. Click Done.

Configuring Protocol Settings

The Protocol Settings screen provides the launching point for configuring bindings, partner endpoints, and other settings needed for the selected SAML profiles (if you are using SAML—see “Choosing SAML Profiles” on page 216). The screen also displays configured information.

(For WS-Federation, the configuration of bindings is not applicable.)

<table>
<thead>
<tr>
<th>Adapter Contract</th>
<th>Source</th>
<th>Value</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>First Name</td>
<td>Assertion</td>
<td>firstname</td>
<td>None available</td>
</tr>
<tr>
<td>Last Name</td>
<td>Assertion</td>
<td>lastname</td>
<td>None available</td>
</tr>
<tr>
<td>Email Address</td>
<td>Assertion</td>
<td>email</td>
<td>None available</td>
</tr>
</tbody>
</table>
To configure Protocol Settings, you need to know:

- For SP-initiated SSO profiles, the URL(s) of your IdP's Single Sign-on Service(s).
- For SLO profiles, the URL(s) of your IdP's Single Logout Service(s).
- When artifact is an allowable inbound binding, the URL of your IdP's Artifact Resolution Service(s).
- The transport configurations (bindings) that you will use to send and receive data for SSO/SLO connections.
- Digital signature policies and certification requirements to which you and your connection partner have agreed.
- XML encryption policies to which you and your connection partner have agreed.

To continue, click Configure Protocol Settings.

**Important:** After modifying any settings, you must click Save on the Protocol Settings screen.

### Specifying SSO Service URLs (SAML)

At this step for SAML 2.0 connections, you associate bindings to the endpoints where your IdP wants PingFederate to send authentication requests when SSO is initiated at your site.

For SAML 1.x, only one endpoint is allowed, and the binding selection is not required.

This configuration applies only to the SP-initiated SSO Profile (see “Configuring SP-Initiated SSO” on page 217).
Chapter 6
Service Provider SSO Configuration

Field Descriptions

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Binding (SAML 2.0)</td>
<td>The transport type agreed upon by you and your partner: Artifact, POST, or Redirect.</td>
</tr>
<tr>
<td>Endpoint URL</td>
<td>The location where your IdP receives SSO messages.</td>
</tr>
</tbody>
</table>

To reach this screen for editing:
1. Click a connection name on the Main Menu.
   Click Manage All IdP, if needed, to see a full list of connections.
2. Click Browser SSO under the IdP Connection tab.
3. Click Configure Browser SSO.
5. Click Configure Protocol Settings.
6. Click SSO Service URL(s) on the Summary screen.
   If this step is not displayed, you have not selected SP-initiated SSO (see “Choosing SAML Profiles” on page 216).

To define an Endpoint URL:
1. If you are using SAML 2.0, select the Binding your partner specifies for the Endpoint.
2. Enter the fully qualified Endpoint URL or a relative path if you have defined a base URL (see “General Information” on page 212).
3. If you are using SAML 2.0, click Add.
4. If your partner has additional SSO endpoints established under SAML 2.0, repeat the steps above.

Specifying a Service URL (WS-Federation)
The Service URL is the WS-Federation endpoint of your IdP partner. This endpoint is where you send RST (Request for Security Token) and SLO messages.
To reach this screen for editing:
1. Click a connection name on the Main Menu.
   Click Manage All IdP, if needed, to see a full list of connections.
2. Click Browser SSO under the IdP Connection tab.
3. Click Configure Browser SSO.
4. Click Protocol Settings under the Browser SSO tab.
5. Click Configure Protocol Settings.
6. Click Service URL on the Summary screen.

Specifying SLO Service URLs (SAML 2.0)

At this step you associate bindings to the endpoints where your IdP receives logout requests when SLO is initiated at your site and where you send SLO responses when you receive SLO requests from the IdP.

This step applies only for SAML 2.0 connections when you have selected an SLO profile (see “Choosing SAML Profiles” on page 216).
Field Descriptions

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Binding</td>
<td>The transport type agreed upon by you and your partner: Artifact, POST, Redirect, or SOAP.</td>
</tr>
<tr>
<td>Endpoint URL</td>
<td>The location where your IdP receives SLO request messages.</td>
</tr>
<tr>
<td>Response URL</td>
<td>(Optional) The location where the IdP receives logout responses. Use this endpoint when you are part of a chain of session participants.</td>
</tr>
</tbody>
</table>

To reach this screen for editing:
1. Click a connection name on the Main Menu.  
   Click Manage All IdP, if needed, to see a full list of connections.
2. Click Browser SSO under the IdP Connection tab.
3. Click Configure Browser SSO.
5. Click Configure Protocol Settings.
6. Click SLO Service on the Summary screen.

To define an Endpoint URL:
1. Select the Binding your partner specifies for the Endpoint.
2. Enter the fully qualified Endpoint URL or a relative path if you have defined a base URL (see “General Information” on page 212).
3. (Optional) Enter the Response URL or a relative path and click Add.
4. If your partner provides additional endpoints for SLO, repeat the steps above.

Choosing Allowable SAML Bindings (SAML)

At this step you configure binding(s) that the IdP will use to send SAML assertions or SLO messages (under SAML 2.0) to your PingFederate server.

This configuration applies to all profile types (see “Choosing SAML Profiles” on page 216). You and your partner can agree to standardize on one binding type or select different bindings for different profile scenarios.
Managing IdP Connections

To reach this screen for editing:

1. Click a connection name on the Main Menu.
   Click Manage All IdP, if needed, to see a full list of connections.
2. Click Browser SSO under the IdP Connection tab.
3. Click Configure Browser SSO.
5. Click Configure Protocol Settings.
6. Click Allowable SAML Bindings on the Summary screen.

To define binding requirements for this connection:

- Make your selections and click Next (or Done).

Setting an Artifact Lifetime (SAML 2.0)

When you send an artifact to your IdP’s SSO or SLO service, an element in the message indicates how long it should be considered valid.

You can change the default value per your requirements, if needed. Also consider synchronizing clocks between your server and your partner’s SAML gateway server. If clocks are not synchronized, you might need to set the artifact lifetime to a higher value.
To reach this screen for editing:
1. Click a connection name on the Main Menu.
   - Click Manage All IdP, if needed, to see a full list of connections.
2. Click Browser SSO under the IdP Connection tab.
3. Click Configure Browser SSO.
5. Click Configure Protocol Settings.
6. Click Artifact Lifetime on the Summary screen.
   - This step appears only if you have selected the artifact binding for either a SSO or SLO Service at the IdP site.

Specifying Artifact Resolver Locations

This endpoint or group of endpoints is where your server will send back-channel requests based on artifacts. The location or locations are also known under SAML specifications as the Artifact Resolution Service. SAML 2.0 provides for multiple, indexed endpoints for the service.

<table>
<thead>
<tr>
<th>SAML20</th>
<th>Configuring &quot;Demo IdP&quot; IdP Connection</th>
<th>Help</th>
<th>Support</th>
<th>About</th>
<th>Logout (Administrator)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main</td>
<td>IdP Connection</td>
<td>Browser SSO</td>
<td>Protocol Settings</td>
<td></td>
<td></td>
</tr>
<tr>
<td>✓ SSO Service URLs</td>
<td>✓ SLO Service URLs</td>
<td>✓ Allowable SAML Bindings</td>
<td>✓ Artifact Lifetime</td>
<td>✓ Artifact Resolver Locations</td>
<td>✓ Signature Policy</td>
</tr>
</tbody>
</table>

Note that this screen is different for SAML 1.x implementations, for which only one endpoint is allowed.

To reach this screen for editing:
1. Click a connection name on the Main Menu.
   - Click Manage All IdP, if needed, to see a full list of connections.
2. Click Browser SSO under the IdP Connection tab.
3. Click Configure Browser SSO.
5. Click Configure Protocol Settings.
6. Click Artifact Resolver Locations on the Summary screen.
   - If this step does not appear, you do not have Artifact selected under Allowable SAML Bindings.
For a SAML 2.0 connection:

1. Enter a URL on the Artifact Resolver Locations screen and click Add.
   
   The URL must be fully qualified (defining protocol, host, and port) unless you have entered a base URL (see “General Information” on page 212).

   Repeat this step if your IdP supports multiple services. The SAML 2.0 specifications permit multiple artifact resolution services through the use of index numbers, which PingFederate automatically supplies when you add a service. Alternatively, if needed per partner specifications, you may assign these index numbers manually.

   **Note:** When specifying multiple artifact resolution endpoints, each endpoint must share the same transport protocol. That is, if one endpoint uses HTTP, then all must use HTTP. Similarly, if one endpoint uses HTTPS, then all must use HTTPS.

2. Click Next.

For a SAML 1.x connection:

1. Enter the Endpoint on the Artifact Resolution Location screen.
   
   The URL must be fully qualified (defining transport protocol, host, and port) unless you have entered a base URL (see “General Information” on page 212).

2. (Optional) Enter your partner’s Source ID.
   
   The Source ID is usually a generated value based on a federation partner’s Connection ID; the SP will correctly generate the Source ID. If that is the case for this partner, then leave this field blank. If your partner uses a Source ID that is not based on the Issuer ID, then enter the Source ID supplied by your IdP partner.

3. Click Next.

Configuring Signature Policy

The Signature Policy screen provides options controlling how digital signatures are used for SSO Internet messaging. The choices made on this screen depend on your partner agreement (see “Digital Signing Policy Coordination” on page 12).

Digital signing is required for SAML Response messages sent from the IdP via POST (or Redirect for SAML 2.0). Optionally, SSO authentication requests from the SP (SP-initiated SSO) may also be signed to enforce security. (This option appears only for SAML 2.0 connections and only if you have enabled SP-initiated SSO using the POST or redirect bindings.)

The assertions inside SAML Responses may be also be signed. When you make this choice, only the assertion portion of the Response is signed, not the complete Response. (This is the only option that appears for SAML 1.x connections.)
Chapter 6
Service Provider SSO Configuration

To choose one or more enhancement options, select the second button, make your selection(s), and click Next.

Otherwise, select the first option (if not already selected) and click Next.

Configuring XML Encryption Policy (SAML 2.0)

For SAML 2.0 configurations, in addition to using signed assertions to ensure authenticity, you and your partner may also agree to encrypt all or part of an assertion to improve privacy. This feature is commonly used if the assertion might pass through an intermediary (such as a user's browser) and HTTPS is not used.

If the name identifier (or SAML_SUBJECT) of an assertion is encrypted, you and your partner may also want to encrypt the identifier in subsequent single-logout messages (if you are using an SLO profile).

Note that “The entire assertion” selection on the Encryption Policy screen includes the SAML_SUBJECT and all attributes.
To reach this screen for editing:
1. Click a connection name on the Main Menu.
   Click Manage All IdP, if needed, to see a full list of connections.
2. Click Browser SSO under the IdP Connection tab.
3. Click Configure Browser SSO.
4. Click Protocol Settings under the Browser SSO tab.
5. Click Configure Protocol Settings.

To define XML encryption:
1. Select Allow encrypted SAML Assertions and SLO messages.
2. Choose whether this IdP partner will encrypt the entire assertion, the
   SAML_SUBJECT, one or more other attributes, or some combination.
3. If your partner is encrypting the name-identifier attribute, use the checkboxes
   near the bottom of the screen to indicate whether you will encrypt this attribute
   in outbound SLO messages and/or allow its encryption for inbound messages.
4. Click Next or Done.

To disable previously configured XML encryption selections:
1. Select None and then Done.
2. Click Save on the Browser SSO screen.

Editing and Saving Protocol Settings
On the Summary screen you can review or edit your Protocol Settings.

Important: When you finish editing existing settings, be sure to
click Done on the Summary screen and then Save on the Protocol
Settings screen. For a new connection, click Done and then click
Next on the Protocol Settings screen. Save the entire connection on
the Activation & Summary screen (see “Connection Activation and
Summary” on page 265).

To reconfigure saved settings:
1. Click the heading over the information you want to change.
2. Click Done on the screen containing your change.
   If you need to make dependent or other changes, do so and continue by clicking
   Done until you reach the Protocol Settings screen.
3. Click Save on the Protocol Settings screen.
Chapter 6
Service Provider SSO Configuration

Editing and Saving Browser SSO Settings

On the Summary screen for Browser SSO, you can review or edit your SSO configuration.

Important: When you finish editing existing settings, be sure to click Done on the Summary screen and then Save on the Browser SSO screen. For a new connection, click Done and then click Next on the Browser SSO screen. Save the entire connection on the Activation & Summary screen (see “Connection Activation and Summary” on page 265).

To reconfigure saved settings:
1. Click the heading over the information you want to change.
2. Click Done on the screen containing your change.
   If you need to make dependent or other changes, do so and continue by clicking Done until you reach the Browser SSO screen.
3. Click Save on the Browser SSO screen.

User Provisioning

PingFederate's Express Provisioning allows SPs to create user accounts “on the fly” during SSO events, based on attributes received from IdPs (see “Express Provisioning” on page 18). An SP can also use the feature to update existing user records.

This configuration task is presented in the administrative console only when User Provisioning is selected as an option (see “Choosing Connection Options” on page 210).

To continue, click Configure User Provisioning.
Choosing an Event Trigger

On the Event Trigger screen, choose whether PingFederate initiates user provisioning only when the user identifier is new or every time your site receives a SAML assertion. In the latter case, an existing user account is always updated with incoming attributes.

Selecting Attribute Sources (SAML 2.0)

For SAML 2.0 connections, the server can be configured to use only assertion attributes for user provisioning or to retrieve more attributes from the IdP in a follow-on Attribute Query transaction (see “Attribute Query and XASP” in the “Supported Standards” chapter of Getting Started). The User Attributes screen displays the attributes expected in the assertion from this IdP (see “Creating an Attribute Contract” on page 220).

**Note:** Attribute Query is a SAML 2.0 profile. For SAML 1.x and WS-Federation connections, this screen is not presented; PingFederate uses only attributes from the assertion for user provisioning.

If you and your IdP partner have agreed to use the Attribute Query profile for provisioning, select that option before leaving this screen.

You will configure the Attribute Query profile later in the task flow, if you have not already done so (see “Configuring the Attribute Query Profile” on page 252).
Identifying the User Repository

PingFederate's Express Provisioning currently supports LDAP v3-compliant user stores and the JDBC-compliant Microsoft SQL Server 2005.

Note: We recommend using the latest version of the Microsoft SQL Server JDBC Driver (version 4 or higher), which is compatible with current and recent JDK versions.

Choose the data store on the User Repository screen.

If the correct data store is not shown in the drop-down list, then PingFederate is not yet configured to access the store (see “Managing Data Stores” on page 72).

▶ If you are using an LDAP store, refer to the sections immediately following:
  • “Specifying an LDAP User-Record Location”
  • “Specifying an LDAP Filter” on page 247
  • “Identifying Provisioning Attributes for LDAP” on page 248

▶ If you are using MSSQL, skip to these sections:
  • “Specifying a Database User-Record Location” on page 248
  • “Specifying a Unique-ID Database Column” on page 249

Specifying an LDAP User-Record Location

After choosing a data store, indicate where in the store PingFederate should write new user records or update existing ones (see “Choosing an Event Trigger” on page 245). Start by specifying where user records are located in your data store.
Specifying an LDAP Filter

On the Unique ID screen, create an LDAP filter to identify user accounts to be provisioned (or updated) during SSO events. PingFederate uses this expression in conjunction with the Base DN (see the previous section) to locate existing account records and to add new ones.

The filter is in the form:

attribute=${value}

Note that the statement must not be enclosed in parentheses, unlike filters used to retrieve LDAP attributes for adapter mapping (see “Configuring an LDAP Filter” on page 231).

The left-side variable is an attribute in your user data store—click the link near the left corner of the screen to see a list of available attributes. The right side of the filter generally uses one or more attribute values passed in from the assertions (see “Creating an Attribute Contract” on page 220). Variables for these attributes, including the correct syntax, are listed under Assertion Values.
Identifying Provisioning Attributes for LDAP

On the Attributes screen, select the data-store attributes to be provisioned.

**Tip:** Multiple-value IdP attributes are handled automatically: when you map a multi-value assertion attribute to an LDAP attribute, each value is stored separately for the LDAP attribute name. If you need to provide additional values for particular attributes, add the same attribute name to this list multiple times. You can then map the additional values on the Attribute Fulfillment screen (see the next section).

To select attributes:

1. Choose a Root Object Class and an Attribute from the drop-down lists and click **Add Attribute**.
   Repeat this step for the same attribute if needed (see “Tip” above).

2. Repeat the previous step for each attribute requiring provisioning.

**Important:** For the Sun Directory Server or Active Directory, the attribute `objectClass` must be among attributes added—select `<Show All Attributes>` under Root Object Class to locate and add this attribute.

Specifying a Database User-Record Location

For database provisioning, indicate where PingFederate should write new user records or update existing ones (see “Choosing an Event Trigger” on page 245).
Field Descriptions

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schema</td>
<td>Defines the table structures that store information within the database.</td>
</tr>
<tr>
<td>Table</td>
<td>Select the name of the database table that contains user records. The selection should be one.</td>
</tr>
<tr>
<td>Columns to fulfill on INSERT</td>
<td>For the selected table, all attributes and their data types are displayed. All attributes must be mapped for the database insertion to succeed, although a null entry may be used for optional attributes (see “Mapping Attributes to User Accounts” on page 250).</td>
</tr>
</tbody>
</table>

Specifying a Unique-ID Database Column

PingFederate uses the column specified on this screen to check whether a user record already exists for the incoming assertion.
Select a column that represents a unique characteristic about the database entry for a particular user (email, for example).

**Mapping Attributes to User Accounts**

The Attribute Fulfillment screen provides a means of mapping the values of incoming attributes into local account attributes. You can also provide values of your own for any attributes—either as hard-coded text or derived values based on assertion attributes.

For JDBC provisioning, this screen also provides a means of testing the insertion of attribute values into the database.

**Map attributes from one of these Sources:**

- ** Assertion  
  Values are contained in the assertions from this IdP. When you make this selection, the associated Value drop-down list is populated by the attribute contract (see “Creating an Attribute Contract” on page 220).**

- ** Attribute Query  
  This choice appears only if you have chosen to use the Attribute Query profile for provisioning (see “Selecting Attribute Sources (SAML 2.0)” on page 245).  

To map an attribute-query value, use this syntax:

```text
${query_attribute}
```

You can also combine attribute-query values with references to attributes in the attribute contract. For example:

```text
${query_attribute}+${attribute}
```

References to attributes not contained in the attribute contract result in an Attribute Query back to the IdP partner.
Managing IdP Connections

- **Expression (when enabled)**
  This option provides more complex mapping capabilities—for example, transforming incoming values into different formats (see “Using Attribute Mapping Expressions” on page 353). All of the variables available for text entries (see below) are also available for expressions.

- **Text**
  The value is what you enter. This can be text only, or you can mix text with references to any of the values from the assertion, using the ${attribute} syntax.

  **Note:** If no entry is required in the JDBC database for an attribute, you can leave the text box blank for character data types or enter null for other types.

**To map attributes:**

1. Choose a Source for each Target attribute (see descriptions of each Source type above).

   **Tip:** For LDAP mapping, choose Text as the Source for the objectClass attribute.

2. Choose (or enter) a Value for each Attribute.
   All values must be mapped. However, you can enter null as a text value for any optional JDBC attributes (or leave the text box blank for optional string attributes).

   **Tip:** For Active Directory, enter user in the text box for objectClass. For the Sun Directory Server, enter inetOrgPerson.

3. Click **Done**.

**To test the insertion of attributes into a JDBC database:**

1. Click the **Test insert into . . .** link.

2. For each Target Attribute, enter text into the boxes under Test Values and click **Test Insert**.
   If the test is successful, a confirmation is displayed along with the values inserted.

3. Unless you wish to keep test values in the database, click **Roll Back All Test Inserts**.

4. When finished, click **Next** or **Done**; or click the link **Return to Attribute Fulfillment** to continue or change any mapping.

**Error Handling**

If user provisioning fails for any reason during SSO events, you can choose to stop the SSO or continue the process by passing assertion attributes on to your application (via the SP adapter configuration—see “Configuring Adapter Mapping and User Lookup” on page 222).
When SSO is aborted, the user is redirected to an error page, and the failure is written to the log and added to the Transaction Failure Count at the bottom of the administrative console.

**Using the Provisioning Summary Screen**

The Summary screen provides an overview of your provisioning configuration.

- When you are finished with a new configuration, click **Done** and then **Save** on the User Provisioning screen.

**To change the configuration:**

1. Click the heading over the information you want to change.
2. Click **Done** on the screen containing your change.
   - If you need to make additional changes, do so and continue by clicking **Done** until you reach the User Provisioning screen.
3. Click **Save** on the User Provisioning screen.

**Configuring the Attribute Query Profile**

At the Attribute Query step you configure your connection to request user attributes from your partner IdP, if you have chosen this option (see “Choosing Connection Options” on page 210). Attribute queries are not dependent on single sign-on but may be used independently or in conjunction with Browser SSO or provisioning to provide flexibility in how a user authenticates with SP applications (see “Attribute Query and XASP” in the “Supported Standards” chapter of *Getting Started*).

**Setting the Attribute Authority Service URL**

*Attribute Authority* is the term used to refer to an IdP that provides user attributes to an *Attribute Requester* (your SP site). The Attribute Authority Service URL corresponds to the endpoint location where Attribute Query requests are received by your IdP partner (see “Attribute Query and XASP” in the “Supported Standards” chapter of *Getting Started*).
Managing IdP Connections

To configure the URL:

- Enter the fully qualified URL or a relative path if you have defined a base URL (see “General Information” on page 212).

Mapping Attribute Names

If the application at your site uses different names for user attributes than the names defined by the Attribute Authority, then you need to map them on this screen. When the SP receives a request from a local application to send an Attribute Query to this Attribute Authority partner, the requested user attributes are replaced with the names mapped here.

This information must be predetermined in your agreement with this connection partner.

To map attributes:

1. Enter the Local Name and Remote Name of an attribute and click Add.
   - Repeat this step for all attributes requiring mapping.
2. Click Next.

To edit a mapping:

1. Click Edit under Action for the mapping.
2. Make your change(s) and click Update.

Note: If you change your mind, ensure that you click the Cancel link in the Actions column, not the Cancel button, which discards any other changes you might have made in this configuration.

3. Click Done and then Save on the Attribute Query screen.
Chapter 6
Service Provider SSO Configuration

Specifying Security Policy

This screen allows you to specify the digital signing and encryption policy to which you and your partner have agreed. These selections will trigger requirements for setting up Credentials (see “Configuring Credentials” on page 254).

This screen also allows you to mask incoming attribute values in log files (see “Attribute Masking” on page 9). When you enable this selection, all user attributes returned from this IdP are masked.

To configure attribute-query security policy for this partner:

1. Check or clear the check boxes and click Next or Done.

Editing and Saving Attribute Query Configurations

On the Summary screen you can review the Attribute Query configuration.

To reconfigure saved profiles:

1. Click the heading over the information you want to change.
2. Click Done on the screen containing your change.
   If you need to make additional changes, do so and continue by clicking Done until you reach the Attribute Query screen.
3. Click Save on the Attribute Query screen.

Configuring Credentials

The Credentials screen presents a list of possible security requirements you might need, depending on the federation protocol you are using and the choices you have made.

Your connection configuration may involve any or all of the following:

- Configuring Back-Channel Authentication
- Configuring Digital Signature Settings
- Configuring Signature Verification Settings
- Selecting an Encryption Certificate
- Selecting a Decryption Key
To configure or modify credentials, click **Configure Credentials**.

### Configuring Back-Channel Authentication

When you configure a profile for the **inbound** artifact binding or the **outbound** SOAP binding, you must specify back-channel authentication information for sending SOAP messages or artifact resolution requests to your partner IdP.

Similarly, if you send artifacts or SOAP messages to your partner IdP, then you must configure SOAP authentication requirements for receiving SOAP responses or artifact resolution requests from your partner.

This step also applies to attribute-request configurations, since this profile always uses the SOAP back channel (see “Choosing SAML Profiles” on page 216).

- To configure or modify credentials, click **Configure Credentials**.

### Send to your partner:
- SOAP SLO messages
- Artifact resolution requests
- Attribute Query requests

### Receive from your partner:
- SOAP SLO messages
- Artifact resolution requests
- Attribute Query requests

**Note:** A yellow triangle next to a listing indicates that you have not completely configured back-channel authentication requirements.
To reach this screen for editing:
1. Click a connection name on the Main Menu.
   Click Manage All IdP, if needed, to see a full list of connections.
2. Click Credentials under the IdP Connection tab.
3. Click Configure Credentials.
   If the Back-Channel Authentication step is not shown, then it is not applicable to your configuration—you are not using the Attribute Query profile and have not configured any profiles to use the artifact or SOAP bindings.

To configure back-channel authentication requirements for sending SOAP messages:
1. On the Back-Channel Configuration screen, click the Configure link to the right of the list of messages to be sent to your partner.
2. Make one or more selections on the Outbound SOAP Authentication Type screen:
   • Basic — you will enter SOAP Basic credentials on a later screen.
   • SSL Client Certificate — you will specify the certificate on a later screen.
     This option is enabled only if you have specified an endpoint that uses SSL.
   • Use Digital Signatures . . . — you will sign the message.
     You will be asked to select a signing certificate on a later screen.

For SAML 2.0, these options may be used in any combination or independently. For SAML 1.x, you must use either Basic or SSL authentication; digital signing may be added to ensure message integrity.

By default, PingFederate validates your partner’s SSL server certificate—verifying that the certificate chain is rooted by a trusted Certificate Authority and that the hostname matches the certificate’s Common Name. Clear the associated checkbox if you do not want this validation to occur.
3. (Optional) On the Outbound SOAP Authentication Type screen, select the checkbox requiring a valid certificate chain for your partner's SSL certificate.
   Make this selection only if you and your partner have agreed that the chain of authority is required for SSL federation transactions.
4. Click Next.
5. If you chose Basic at Step 2, enter the SOAP Username and Password to use for this partner under Basic SOAP Authentication.
   You must obtain these credentials from your partner.
6. If you are using an SSL certificate, select the certificate under SSL Authentication Certificate and click Next.
   If you have not yet created or imported the client SSL certificate you need into PingFederate, click Manage Certificates (see “SSL Client Keys & Certificates” on page 91). You will need to export the certificate (only) and send it to your partner.
7. On the Summary screen, click Done.

To configure back-channel authentication requirements for receiving SOAP messages:
1. On the Back-Channel Configuration screen, click the Configure link to the right of the list of messages to be received from your partner.
2. Select one or more options on the Inbound SOAP Authentication Type screen:
Managing IdP Connections

- Basic — Enter the logon username and password your partner will use on the next screen.
- SSL Certificate — Specify certificate verification information on a later screen.
- Use Digital Signatures . . . — Incoming messages must be signed.

You will be asked to select a signature verification certificate on a later screen.

For SAML 2.0, these options may be used in any combination or independently. For SAML 1.x, you must use either Basic or SSL authentication; digital signing may be added to ensure message integrity.

3. Click Next.

4. If you chose Basic at Step 2, enter the SOAP Username and Password under Basic SOAP Authentication.

Important: If you are configuring more than one connection that uses the artifact or SOAP profile, you must ensure that the Username is unique for each connection.

5. If you are using an SSL certificate, select Anchored or Unanchored under Certificate Verification Method.
   - Anchored — The certificate must be signed by a trusted Certificate Authority, and the CA’s certificate must be imported into the PingFederate Trusted CA store (see “Trusted Certificate Authorities” on page 87).
   - Unanchored — The certificate is self-signed or you wish to trust a specified certificate.

Note: When anchored certificates are used between partners, certificates may be changed without sending the update to your partner. If the certificate is unanchored, any changes must be promulgated.

6. Click Next.

7. If you chose anchored SSL certificate verification at Step 5, enter the Subject DN and click Next.

Tip: If you have not yet defined the certificate in PingFederate or you do not know the DN, return to the previous screen and check Unanchored. Then click Next and click Manage Certificates on the SSL Verification Certificate screen to import the certificate, if needed, or to view its DN.

8. If you chose unanchored SSL certificate verification at Step 5, select the certificate you will use to validate the SSL connection.

   If you have not yet imported the certificate into PingFederate, click Manage Certificates.

9. Click Next.

10. On the Summary screen, click Done.
Configuring Digital Signature Settings

This step defines the private key you will use to sign SSO authentication or attribute requests (optionally) or SAML 2.0 SLO messages for this IdP. In addition, the step allows you to include “Key Info” with the XML message if you and your partner have agreed to this option.

Digital signing applies to SP-initiated SSO under SAML 2.0, when specified by your partner agreement, and to either SLO profile (see “Choosing SAML Profiles” on page 216) using the POST or redirect bindings. The step also applies if you are configuring an Attribute Query profile and have specified that you will sign attribute requests (see “Specifying Security Policy” on page 254).

The step is not required for SAML 1.x IdP connections.

To reach this screen for editing:

1. Click a connection name on the Main Menu.
2. Click Manage All IdP, if needed, to see a full list of connections.
3. Click Credentials under the IdP Connection tab.
4. Click Configure Credentials.
5. Click Digital Signature Settings on the Summary screen.

To specify a certificate:

1. Select the certificate from the drop-down list.
2. (Optional) If you have agreed to send your public key with the SAML message, click the checkbox to include the certificate.

Configuring Signature Verification Settings

Under SAML 2.0 specifications, when your site receives any SAML 2.0 messages via the POST or Redirect bindings, the messages must be digitally signed. Signing is also
always required for the SAML 1.0 POST binding and for WS-Federation assertions, as well as incoming SAML 1.1 or 2.0 tokens for WS-Trust STS processing.

Depending on your agreement with this IdP, SSO assertions, SAML 2.0 artifacts, or SOAP messages might also require signatures.

Whenever signatures are required, PingFederate provides a choice of trust models, including an option to use anchored signature-verification certificates embedded in incoming messages (see “Trust Models” on page 12). When this option is chosen in Signature Verification Settings, you must provide the Subject DN for embedded certificates coming from this partner, and the Issuer CA certificate must be part of the PingFederate trusted store (see “Trusted Certificate Authorities” on page 87).

Alternatively, you may choose to use unanchored certificates, in which case you must import your partner’s public-key certificate during this configuration (or select it if it is already imported). To prevent any interruption of service due to an expired certificate, you can ask your partner for a new certificate in advance and import it as backup.

To continue, click Manage Signature Verification Settings.

To reach this screen for editing:
1. Click a connection name on the Main Menu.
   Click Manage All IdP, if needed, to see a full list of connections.
2. Click Credentials under the IdP Connection tab.
3. Click Configure Credentials.
4. Click Signature Verification Settings on the Summary screen.
   If this step does not appear, then your configuration does not require verification settings.

Choosing a Trust Model

This screen allows you to choose the Trust Model you want to use for signature verification (see “Trust Models” on page 12).
Depending on the selection, the next step in this task varies:

- **For Anchored**, the next step is to enter the Subject DN for your partner's certificate (see the next section, "Anchored Certificates: Specifying a Subject DN").

**Important:** If you are using the Redirect binding for SLO, you cannot use anchored certificates because SAML 2.0 does not permit certificates to be included using this transport method.

- **For Unanchored**, the next step is to import your partner's certificate (see “Selecting Unanchored Certificates” on page 261).

**To reach this screen for editing:**

1. Click a connection name on the Main Menu.
   
   Click **Manage All IdP**, if needed, to see a full list of connections.

2. Click **Credentials** under the IdP Connection tab.

3. Click **Configure Credentials**.

4. Click **Signature Verification Settings** on the Summary screen.
   
   If this step does not appear, then your configuration does not require verification settings.

5. Click **Manage Signature Verification Settings**.

6. Click **Trust Model** on the Summary screen.

**Anchored Certificates: Specifying a Subject DN**

When you choose to use an anchored certificate for signature verification, incoming SAML messages must contain the partner's verification certificate (see “Trust Models” on page 12). PingFederate verifies that the Issuer CA is trusted and checks to see that the embedded certificate's Subject DN matches the one specified on this screen. If so, PingFederate uses that certificate to verify the message signature.
Managing IdP Connections

To extract the Subject DN from a certificate:
1. Click browse to select your IdP partner’s public certificate.
2. Click Extract.

To reach this screen for editing:
1. Click a connection name on the Main Menu.
   Click Manage All IdP, if needed, to see a full list of connections.
2. Click Credentials under the IdP Connection tab.
3. Click Configure Credentials.
4. Click Signature Verification Settings on the Summary screen.
   If this step does not appear, then your configuration does not require verification settings.
5. Click Manage Signature Verification Settings.
6. Click Certificate Subject DN on the Summary screen.

Selecting Unanchored Certificates

On the Signature Verification Certificate screen, you identify your partner’s imported public certificate and, optionally, a secondary certificate to use when the first expires (see “Trust Models” on page 12).
Chapter 6
Service Provider SSO Configuration

To reach this screen for editing:

1. Click a connection name on the Main Menu.
   Click Manage All IdP, if needed, to see a full list of connections.

2. Click Credentials under the IdP Connection tab.

3. Click Configure Credentials.

4. Click Signature Verification Settings on the Summary screen.
   If this step does not appear, then your configuration does not require verification settings.

5. Click Manage Signature Verification Settings.

6. Click Signature Verification Certificate on the Summary screen.

To specify a verification certificate:

1. Select the certificate from the drop-down list.
   If you have not yet imported the certificate into PingFederate, click Manage Certificates.

2. (Optional) Select a Secondary certificate for backup.
   Use this field if your partner has sent you a new certificate to replace one that is ready to expire. The server will automatically verify against the secondary certificate when the primary one expires.

Selecting an Encryption Certificate

If SAML_SUBJECT is encrypted, either by itself or as part of a whole assertion, then all references to this name identifier in SAML 2.0 SLO requests from your site may also be encrypted (if the connection uses SP-initiated SLO). For more information, see “Configuring XML Encryption Policy (SAML 2.0)” on page 242.

To enable this XML encryption, you must identify an encryption certificate for this partner.

You must also choose a certificate if encryption of the Name Identifier is required for an Attribute Request profile (see “Specifying Security Policy” on page 254).
Managing IdP Connections

To reach this screen for editing:
1. Click a connection name on the Main Menu.
   Click Manage All IdP, if needed, to see a full list of connections.
2. Click Credentials under the IdP Connection tab.
3. Click Configure Credentials.
   If this step is not present, then you have either not configured this connection to use the SP-initiated SLO profile (see “Choosing SAML Profiles” on page 216) or you have chosen not to encrypt the assertion or the SAML_SUBJECT (see “Configuring XML Encryption Policy (SAML 2.0)” on page 242).

To identify the encryption certificate:
1. (Optional) Change the default settings under Block Encryption Algorithm and/or Key Transport Algorithm.
   Due to import restrictions, the standard JRE distribution supports strong but not unlimited encryption. To use the strongest AES encryption, when permissible, download and install the appropriate version of “Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files” from the Sun download Web site (http://java.sun.com/javase/downloads).
   For more information about XML block encryption and key transport algorithms, see the “XML Encryption Syntax and Processing W3C Recommendation” (http://www.w3.org/TR/xmlenc-core).
2. From the drop-down list, select the applicable certificate and click Next.
   If the certificate is not in the list, click Manage Certificates to import it.

Note: If you have already imported a signature verification certificate for this partner, you can reuse it for XML decryption as long as it is an RSA certificate.
Selecting a Decryption Key

As part of XML encryption, you must identify a signing certificate and key for PingFederate to use to decrypt incoming assertions or assertion elements (see “Configuring XML Encryption Policy (SAML 2.0)” on page 242).

To reach this screen for editing:

1. Click a connection name on the Main Menu.
   - Click Manage All IdP, if needed, to see a full list of connections.
2. Click Credentials under the IdP Connection tab.
3. Click Configure Credentials.
4. Click Select XML Decryption Key.
   - If this step is not present, you have not chosen to require encryption of all or part of the SAML assertion (see “Configuring XML Encryption Policy (SAML 2.0)” on page 242).

To identify the decryption key:

- From the drop-down list, select the applicable certificate and click Next.
  - If the certificate is not in the list, click Manage Certificates to import it (see “Digital Signing and Decryption Keys & Certificates” on page 93).

**Note:** If you have imported a certificate for this partner to use for digital signing, you can reuse it for XML decryption as long as it is an RSA certificate.

Editing and Saving Credential Configurations

From the Summary screen you can review or edit your credentials configuration.

**Important:** When you finish editing existing settings, you must click Done on the Summary screen and then Save on the Credentials screen. For a new connection, click Done and then click Next on the Credentials screen. Save the entire connection on the Activation screen (see “Connection Activation and Summary” next).
Connection Activation and Summary

When you finish setting up a connection, you may choose to activate it immediately.

**Important:** Regardless of whether you choose to activate a new connection now or later, you must click Save on the Summary screen for a new connection if you want to keep the configuration.

You can deactivate a connection at any time (for maintenance, for example). When a connection is inactive, all SSO or SLO transactions to or from this partner are disabled, as well as access to the WS-Trust STS for Web Service Providers associated with this connection.

**Tip:** The SSO Application Endpoint near the top of the Summary screen is an example URL that webmasters or Web application developers at your site might use to invoke SSO for the connection. For details about SSO and other server endpoints, including optional query parameters, see “Application Endpoints” on page 335.

**To change a Connection Status:**

- Select either Active or Inactive and then click Save.

**To modify a connection setting:**

1. If you know which step needs to be modified, click its link under the IdP Connection tab.
   - If you do not know where to change the setting, locate the currently configured data under one of the summary headings and then click the subheading above the data.

2. Change the information on the step screen and click Save, if available.
   - If Save is not available, you are in the middle of a task (see “About Tasks and Steps” in the “Console Navigation” chapter of Getting Started); click Next or Done until you reach a screen containing a Save button. Then click Save and continue as needed until you return to the Main Menu.
   - If your modification requires related configuration changes, PingFederate provides error messages indicating the necessary steps and then guides you to the related screens (unless you click Cancel).

**Important:** Be sure to click Save whenever that button appears, if you want to keep your changes.

Configuring IdP Auto-Connect

When your IdP partner is also using PingFederate 5 or higher (or is otherwise able to provide interoperable SAML 2.0 metadata via HTTP on demand), you may choose to use Auto-Connect for that partner (see “Using Auto-Connect” on page 15). This configuration can be shared among an unlimited number of SAML 2.0 partners.

**Note:** You enable the SAML 2.0 Auto-Connect profile under System Settings (see “Choosing Roles and Protocols” on page 64).
Once Auto-Connect is enabled on your PingFederate server, you complete the configuration from the Main Menu under My SP Configuration. This configuration entails:

- Setting up a common connection for all Auto-Connect partners
- Establishing a list of IdP partner domains authorized to use the connection

### Configuring the Initial Setup

The basic configuration for IdP Auto-Connect requires only:

- Choosing a signing certificate for authentication requests and other SAML messages
- Configuring user-session creation information

All other partner-connection specifications are handled automatically at runtime.

### Choosing a Signing Certificate

For Auto-Connect runtime processing, authentication requests and SLO messages must be signed, since they are sent over either the POST or redirect bindings (see “SAML 2.0 Profiles” in the “Supported Standards” chapter of Getting Started).

**Note:** The signing certificate is embedded in your server’s Auto-Connect metadata (see “Using Auto-Connect” on page 15); there is no need to exchange certificates with your partners.

You can use the same certificate used for signing metadata (see “Configuring Auto-Connect Metadata Signing” on page 70). If you use a different certificate, ensure that it meets Auto-Connect validation requirements (see “Auto-Connect Security Model” on page 17).

### Configuring User-Session Creation

Configuring user-session creation for Auto-Connect is similar to configuring the same settings for regular partner connections.
Click **Configure User-Session Creation** to continue.

For configuration information, refer to sections under "User-Session Creation" on page 218.

**Note:** Attributes sent from the IdP via Auto-Connect are passed to your applications, regardless of whether they are listed in the attribute contract (see "Creating an Attribute Contract" on page 220).

### Connection Activation and Summary

When you finish configuring your IdP Auto-Connect initial setup, you may choose to activate the common connection immediately on the Activation & Summary screen. (No runtime processing occurs until your partner's Auto-Connect gateway is also established and a user initiates an SSO or SLO event.)

**Important:** Regardless of whether you choose to activate a newly configured connection now or later, you must click **Save** on the Activation & Summary screen if you want to keep the configuration.

You can deactivate the connection at any time (for maintenance, for example). While a connection is inactive, all SSO or SLO transactions to or from Auto-Connect partners are disabled.

**To change a Connection Status:**

- Select Active or Inactive and then click **Save**.

**To modify a setting:**

1. Locate the currently configured setting under one of the summary headings and then click the subheading above the data.

**Note:** Changes made to Auto-Connect settings will be out of sync, temporarily, with metadata caches that any currently active partners might be using. If your connection is in production, you might wish to lower your server's metadata lifetime in advance of making configuration changes (see "Configuring Auto-Connect Metadata Lifetime" on page 71).
2. Change the information and click **Save**, if available.
   If **Save** is not available, additional, dependent changes are required; click **Next**
   or **Done** until you reach a screen containing a **Save** button. Then click **Save**
   and continue as needed until you return to the **Main Menu**.

### Specifying Allowed IdP Domains

This screen provides PingFederate with a list of trusted domain names of your Auto-
Connect partners.

<table>
<thead>
<tr>
<th>Domain Name</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>pingidentity.com</td>
<td>Edit / Delete</td>
</tr>
</tbody>
</table>

Normally, when PingFederate receives an SSO request from a Web application at
your site (see “/sp/startSSO.ping” on page 338), the runtime engine completes the
connection automatically using metadata obtained from a standard, public
location—`http://saml.<domain_name>`.(See “Using Auto-Connect” on
page 15.) Alternatively, if an Auto-Connect partner elects not to use the standard
location, you can supply the applicable URL.

**To add a domain:**

▶ Enter a Domain Name and click **Add**.

**To specify a URL for metadata retrieval:**

1. Click **Advanced View**.
2. Enter the Domain Name if you have not already done so.
3. Enter the Metadata Service URL.
   This entry must be obtained from your Auto-Connect partner.
4. Click **Add**.

**Note:** Once you have added the URL, you cannot return to the
**Basic View** unless you first remove the URL value using the
procedure below.

**To edit an entry:**

1. Click **Edit** under Action for the entry.
2. Make your change and click **Update**.

**To delete an entry:**

▶ Click **Delete** under Action for the entry.
WS-Trust STS Configuration

The PingFederate WS-Trust STS provides security-token validation and creation to extend SSO access to identity-enabled Web Services (see “About WS-Trust STS” on page 2).

The chapter provides instructions for configuring the WS-Trust STS in the administrative console and covers these main topics:

- “Server Settings”
- “IdP Configuration for STS”
- “SP Configuration for STS”

Server Settings

To use the PingFederate WS-Trust STS for partner connections, start by enabling the WS-Trust protocol under Server Settings on the Roles and Protocols screen (see “Choosing Roles and Protocols” on page 64). Once the protocol is enabled, you must identify the STS server with a unique federation identifier for both SAML 2.0 and SAML 1.1 tokens (unless these IDs are already established for corresponding browser-based SSO protocols).

In addition, also under Server Settings, you have the option of requiring authentication globally for access to STS endpoints—(see “Configuring STS Authentication” on page 271).

Enabling the WS-Trust STS

You can enable the WS-Trust STS when you first install PingFederate (see “Running PingFederate for the First Time” in the “Installation” chapter of Getting Started). If you have already installed PingFederate or are upgrading to a new version, use the following procedure.

To enable WS-Trust and make the STS available for partner connections:
1. On the Main Menu under System Settings, click Server Settings.
2. Click **Roles and Protocols** under the Server Settings tab.

3. Select WS-Trust for either the IdP or the SP role, or both, depending on your requirements.

   **Note:** PingFederate fully supports the STS with or without selections of any of the Browser SSO protocols listed above the WS-Trust selections. SAML 1.1 and 2.0 token handling is independent of supported SSO protocols chosen here.

4. Click **Next**.

5. On the Federation Info screen, ensure all required fields are completed.

   **Note:** Identifiers are needed for both SAML 2.0 and SAML 1.x to enable the STS to issue either type of token when requested. If you have not established a federation ID for either of these protocols or do not expect to use one or the other, enter a placeholder (in any format) and return later if needed. (For more information about the fields on this screen, see “Specifying Federation Information” on page 67).

6. (Optional) Click **Next** to go to the WS-Trust STS Settings screen (see the next section, “Configuring STS Authentication”).

7. Click **Save** (on any screen).
Configuring STS Authentication

Server settings may be configured to require that client applications provide credentials to access the PingFederate STS. This is recommended for IdP configurations using the Username Token Processor (available separately).

For other token processors and token generators, trust in the identity of the client is conveyed within the token itself and verified as part of processing. However, administrators may wish to add another layer of security by limiting access to only authenticated clients.

**Note:** When STS authentication is configured, the configuration applies globally for all IdP and SP partner connections configured for STS clients and to all token formats.

To continue, click **Configure WS-Trust STS Authentication**.

### Selecting Authentication Methods

You can choose either HTTP Basic or mutual SSL/TLS authentication (or both) on the Authentication Methods screen. (Note that if both methods are configured, all clients must authenticate using both, not one or the other.)

#### Configuring Basic Authentication

For HTTP Basic authentication, create username/password pairs (“Users”) for all client applications needing access to the STS.
On the HTTP Basic Authentication screen, you can also delete users and update account passwords.

**To add users:**
1. Click **Create User**.
2. On the User Account screen, enter a Username and Password, and confirm the password.
   Passwords must be at least six characters long, containing at least one uppercase, one lowercase, and one numeric character.
3. Click **Done**.
4. Repeat the preceding steps as needed.
5. On the HTTP Basic Authentication screen, click **Next**.
   (If you are also configuring SSL authentication, complete that configuration and click **Next** to reach the Summary screen (see “Configuring Mutual SSL Authentication” on page 272).)
6. On the Summary screen, click **Done**.
7. On the WS-Trust STS Settings screen, click **Save**.

**To update an account password:**
1. Click the Username.
2. On the User Account screen, enter the Current User Password and a New Password, with confirmation.
   Passwords must be at least six characters long, containing at least one uppercase, one lowercase, and one numeric character.
3. Click **Done**.
4. On the HTTP Basic Authentication screen, click **Done**.
5. On the WS-Trust STS Settings screen, click **Save**.

**To delete a user:**
1. Click **Delete** under Action for the Username.
2. Click **Done** (or **Next** for new configuration).
3. Click **Save** when you reach the WS-Trust STS Settings screen.

**Configuring Mutual SSL Authentication**

When SSL authentication is selected on the Authentication Methods screen, the configuration begins on the Mutual SSL Authentication screen.
Choosing Certificate Authentication Options

On the Authentication Options screen, select whether to verify client SSL certificates against a list of Subject Distinguished Names (DNs) or a list of issuer public certificates imported into PingFederate.

**Note:** You can choose both options. However, note that they are not used alternatively at runtime; both validations are applied.

Managing Allowed Subject DNs

On the Allowed Subject DNs screen you can add, edit, or delete Subject DNs for clients allowed to access the PingFederate STS.
Chapter 7
WS-Trust STS Configuration

To add DNs:
1. Enter a valid Subject DN for a partner STS client and click Add.
2. Add other DNs as needed.
3. For a new configuration, click Next or Done to continue.
4. If you are finished with a new or existing configuration, continue clicking Done until you reach the WS-Trust STS Settings screen and then click Save.

To edit DNs:
1. Click Edit under Action for the Subject DN.
2. Make changes and click Update.
3. Click Done.
4. If you are finished with a new or existing configuration, continue clicking Done until you reach the WS-Trust STS Settings screen and then click Save.

To delete entries:
1. Click Delete under Action for the Subject DN.
2. Click Done.
3. If you are finished with a new or existing configuration, continue clicking Done until you reach the WS-Trust STS Settings screen and then click Save.

Managing Allowed Issuer Certificates

When STS access is restricted by issuer certificate, the Allowed Issuer Certificates screen provides a means of maintaining a list of valid certificates.
IdP Configuration for STS

On this screen you can add or remove certificates.

**To add certificates:**
1. Select the certificate from the drop-down list and click **Add**.
   - If the certificate you are looking for is not in the list, click **Manage Certificates** to import it from your file system.
2. Add other certificates as needed.
3. For a new configuration, click **Next** or **Done** to continue.
4. If you are finished with a new or existing configuration, continue clicking **Done** until you reach the WS-Trust STS Settings screen and then click **Save**.

**To delete a certificate from the list:**
1. Click **Remove** under **Action** for the Issuer Certificate.
2. Click **Done**.
3. If you are finished with a new or existing configuration, continue clicking **Done** until you reach the WS-Trust STS Settings screen and then click **Save**.

**Using the Summary Screen**

When you have finished configuring Mutual SSL Authentication, you can review the configuration on the Summary screen. If you need to make any changes, click the heading over the information you want to edit.

- To save a new or modified configuration, click **Done** on successive screens until you reach the WS-Trust STS Settings screen and then click **Save**.

**Using the WS-Trust STS Settings Summary Screen**

When you have finished configuring WS-Trust STS Settings, you can review the configuration on the Summary screen. If you need to make any changes, click the heading over the information you want to edit.

- If you are editing an existing connection, click **Done** and on the WS-Trust STS Settings click **Save**.

**IdP Configuration for STS**

This section covers the IdP configuration for the PingFederate WS-Trust STS, which involves:

- “Configuring Token Processors”
Configuring Token Processors

Token Processors are used to validate incoming tokens and token requests to the STS (see "Token Processors and Generators" on page 3). Token Processors for SAML 2.0 and SAML 1.1 tokens are included with the PingFederate installation. This section provides guidance on configuring "instances" of either of the SAML Token Processors. You must configure at least one processor in order to set up an STS connection.

Only one instance of any token processor may be configured.

Additional Token Processors are available from the Ping Identity Web site (at www.pingidentity.com/products/PingFederate-Token-Translators.cfm).

To begin configuring SAML 1.1 or 2.0 Token Processors:

► On the Main Menu, click Token Processors under Application Integration Settings for My IdP Configuration.

If this link is not shown, ensure that the WS-Trust STS is enabled in Server Settings (see "Enabling the WS-Trust STS" on page 269).

To configure a new token-processor instance:

► Click Create New Instance.

To edit an existing instance:

► Click the Instance Name and click the step you need to change.

To delete an instance:

1. Click Delete next to the Instance Name. (To undo the deletion, click Undelete.)

   Note: This option is available only if the processor instance is not in use for a connection.

2. Click Save to confirm the deletion.

Selecting a Token Processor Type

The first step in creating a SAML token-processor instance is choosing the processor type.
To define an instance:

1. Enter the Instance Name and Instance Id on the Type screen.

2. Select SAML 1.1 Token Processor <version> or SAML 2.0 Token Processor <version> from the drop-down menu.

3. Click Next.

**Configuring a SAML Token Processor Instance**

On the Instance Configuration screen, you may use signing-certificate DN checking to limit the valid signatures and certificates for token requests accepted for this SAML token type. (By default, the STS validates digital signatures using all trusted Certificate Authorities (CAs) imported into PingFederate.)

At minimum on this screen, you must indicate a unique identifier for the PingFederate STS. To be accepted, an incoming SAML token must contain this ID in its <audience> element.
To configure the token-processor instance for certificate validation:

1. Enter a URI for Audience.

   This is the ID for the STS for either SAML 1.1 or SAML 2.0 tokens, depending on which processor you are configuring (see “Specifying Federation Information” on page 67).

2. (Optional) Click the Add a new... link under Action for either Valid Certificate Issuer DNs or Valid Certificate Subject DNs.
   You can use both lists.

   **Important:** When both types of validation are configured, then the certificate used to validate signatures must match an entry in both lists. If only Subject DNs are listed on this screen, then the certificate Issuer DN is not checked and its Subject DN must match one of the entries in the Subject DNs list. If only Issuer DNs are listed here, then the certificate Subject DN is not checked and its Issuer DN must match one of the entries in the Issuer DNs list. If neither Issuer DNs nor Subject DNs are listed, then all certificates are treated as valid for purposes of verification.

3. (Optional) Enter a Valid DN and click Update.

4. (Optional) Repeat the previous steps as needed to add more DNs.
Extending a Processor Contract

Token processors allow administrators to add to a built-in list of user attributes that the processor returns from an incoming token—an extended processor-attribute contract.

To add an attribute:

1. Enter the attribute name in the text box and click Add.

Setting Attribute Masking

On the Token Attributes screen, you can choose to mask attribute values that PingFederate logs from this processor instance at runtime (see “Attribute Masking” on page 9).

To mask an attribute in log files:

1. Under Mask Log Values select the attribute whose value you want to mask.

   If OGNL expressions might be used to map derived values into outgoing tokens and you want those values masked, select the related checkbox under the Attribute list (see “Using Attribute Mapping Expressions” on page 353).

Editing and Saving Processor Instances

From the Summary screen, you can reach processor settings for editing.

To edit the configuration:

1. Click the heading above the information you want to change.
2. Make your changes.
3. Click **Done** on the configuration page and **Save** on the Manage Token Processors screen.

**To save a processor instance:**
1. Click **Done** on the Summary screen.
2. Click **Save** on the Manage Token Processors screen.

### Managing STS Request Parameters

As an option for configuring PingFederate to act as a WS-Trust STS, an administrator can define sets of RST metadata parameters that can be used to map attribute values into issued security tokens. After these request contracts are defined, you can make them available when configuring WS-Trust STS settings for SP-partner connections (see “Selecting a Request Contract” on page 286).

![Configuring My Server](image)

To reach this screen, click **STS Request Parameters** under Application Integration Settings on the Main Menu.

If this link is not present, WS-Trust is not enabled (see “Enabling the WS-Trust STS” on page 269).

To add a new set of request parameters, click **Add New Request Contract**.

To edit an existing contract, click its Contract Name.

### Creating a Request Contract

On the Create Request Contract screen, identify the contract and define parameters that will be available in token requests (as associated with this contract for partner connections—see “Managing STS Request Parameters” on page 280).
Field Descriptions

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contract Name</td>
<td>A descriptive name for the Contract—for example, a Web Service Client or Provider.</td>
</tr>
<tr>
<td>Contract ID</td>
<td>An internal identifier—must be alphanumeric with no spaces.</td>
</tr>
<tr>
<td>Parameters to be provided in the request</td>
<td>A list of request parameters for this Contract (see instructions below).</td>
</tr>
</tbody>
</table>

To add a Parameter:
- Enter the Parameter Name in the text box and click Add.

To modify a Parameter Name:
1. Click Edit under Action for the Parameter Name.
2. Edit the name and click Update.

**Note:** If you change your mind, be sure to click the Cancel link in the Actions column, not the Cancel button, which discards any other changes you might have made.

To delete an attribute:
- Click Delete for the Attribute Name.

**Configuring SP Connections for STS**

You can configure an STS connection to an SP partner either in conjunction with browser-based SSO or independently.
To enable STS for a new connection, or to add the capability to an existing connection:

1. Select the WS-Trust STS option on the Connection Type screen (see “Choosing a Connection Type” on page 114).

   **Note:** Before this option can be selected, the WS-Trust protocol must be enabled in Server Settings (see “Server Settings” on page 269)

2. Select a Default Token Type.
   
   The Default Token Type, either SAML 1.1 or 2.0, is used when a Web Service client does not specify in the token request what token type the STS should issue.

   **Note:** The Default Token Type does not need to match the Protocol indicated on the screen for SSO (when applicable).

When the option is enabled, the configuration starts on the WS-Trust STS screen.

---

### Configuring Protocol Settings

On the Protocol Settings screen, enter a unique identifier for your partner’s Web Service. This identifier corresponds to the element `<AppliesTo>` in Requests for Security Tokens (RSTs).
Also on this screen, options are available for adding signature or encryption protection to outgoing SAML tokens:

- For SAML 1.1, you can choose to generate a public key to be used in conjunction with the “Holder of Key” designation for the assertion’s Subject Confirmation Method (for information about HoK assertions, see, for example, http://docs.sun.com/app/docs/doc/820-1072/6ncp48v45?a=view).

- For SAML 2.0, you can choose to encrypt the assertion.

**Note:** You can make both selections if you expect requests for either type of token. These selections are independent of the Default Token Type selected previously (see “Configuring SP Connections for STS” on page 281)

When you make either (or both) of these selections, you will be asked to choose a signing or XML encryption certificate later in the connection setup, unless required certificates are already in place for an existing browser-based SSO connection. (PingFederate uses the same certificates to handle signing/encryption requirements for both Browser SSO and WS-Trust STS—for more information, see “Configuring Credentials” on page 160.)

### Setting a Token Lifetime

Standards require a window of time during which a security token is considered valid. Each token has a time-stamp XML element as well as elements indicating the allowable lifetime of the token (in minutes) before and after the token time stamp.

---

**SAML 1.1**

- **Generate Key for SAML 1.1 Holder of Key Subject Confirmation Method**
  - *https://pingidentity.com/kvs*

**SAML 2.0**

- **Encrypt SAML 2.0 Assertion**
WS-Trust STS Configuration

Field Descriptions

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minutes Before</td>
<td>The amount of time before the token was issued during which it is to be considered valid.</td>
</tr>
<tr>
<td>Minutes After</td>
<td>The amount of time after the token was issued during which it is to be considered valid.</td>
</tr>
</tbody>
</table>

To change the default times:
▶ (Optional) Edit the desired setting(s) and click Next or Save.

Configuring Token Creation

For the PingFederate STS to issue a security token in response to requests for partner services, you must indicate what user attributes are to be included in the token (the "attribute contract"). The attribute values sent in the token are then derived by mapping those available from the Token Processor you select (see “Attribute Contract Fulfillment” on page 297). As with Browser SSO, the mapping can be augmented using local data stores, variable or constant text, or expressions.

Details of this configuration are handled under the Token Creation task.
Defining an Attribute Contract

An attribute contract is the set of user attributes that a Web Service Client at your site expects to receive in security tokens issued for this connection (see “Attribute Contracts” on page 7). You identify these attributes on this screen.

To reach this screen for editing:

1. Click the connection name on the Main Menu.
2. Click Manage All IdP, if needed, to see a full list of connections.
3. Click WS-Trust STS under the IdP Connection tab.
4. Click Configure WS-Trust STS.
5. Click Configure Token Creation.
6. Click Attribute Contract on the Summary screen.

To add an attribute:

1. Enter the attribute name in the text box.
   Attribute names are case-sensitive and must correspond to the names expected by the requesting WSC.

   **Tip:** You can add a special attribute, SAML_AUTHN_CTX, to indicate to the SP (if required) the type of credentials used to authenticate to the IdP application (see “Authentication Context” in the “Supported Standards” chapter of Getting Started). Map a value for the authentication context on the attribute-mapping screen later in the configuration, from any available attribute source, including the RST if a requested context is specified as a request parameter (see “Attribute Contract Fulfillment” on page 297).

   **Tip:** The Format attribute associated with the NameID element in outgoing SAML tokens may be set when needed by adding an attribute called SAML_NAME_FORMAT. The value of that attribute can then be mapped later (see “Attribute Contract Fulfillment” on page 297).

For information about the NameID elements and applicable URI values, locate the SAML 2.0 specification at oasis-open.org/specs.
2. Click **Add**.

To modify an attribute name:
1. Click **Edit** under Action for the attribute.
2. Make the change and click **Update**.

To delete an attribute:
- Click **Delete** under Action for the attribute.

Selecting a Request Contract

This optional setting allows you to use XML parameters contained in RSTs for token-attribute mapping (see “Managing STS Request Parameters” on page 280).

<table>
<thead>
<tr>
<th>SP Connection</th>
<th>Main</th>
<th>SP Connection</th>
<th>WS-Trust STS</th>
<th>Token Creation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Attribute Contract</strong></td>
<td><strong>Request Contract</strong></td>
<td><strong>IdP Token Processor Mapping</strong></td>
<td><strong>Summary</strong></td>
<td></td>
</tr>
</tbody>
</table>

- If you are not using request parameters, click **Next** to continue.
- To use request parameters, click the checkbox and choose a Request Contract from the drop-down list.

  If the contract you want is not shown, click **Manage STS Request Parameters**.

  When you choose a contract, you will enable an option to select Request from the drop-down Source list on the attribute-mapping screen (see “Attribute Contract Fulfillment” on page 297).

IdP Token Processor Mapping

IdP token processors are responsible for validating incoming security tokens as part of an STS operation (see “Token Processors and Generators” on page 3). A configured and deployed token processor in PingFederate is known as a token processor instance. The same instance may be mapped by multiple connections.

Map one or more IdP token processor instances into each SP connection to handle all the token types that may be received from Web Service Clients associated with this SP partner.

You begin this configuration on the **IdP Token Processor Mapping** screen, where you choose to map instances of IdP token processors. If you have not yet configured an instance of the token processor you intend to use within this SP connection, see “Configuring Token Processors” on page 276.
To reach this screen for editing:

1. Click the connection name on the Main Menu.
   - Click Manage All IdP, if needed, to see a full list of connections.
2. Click WS-Trust STS under the IdP Connection tab.
3. Click Configure WS-Trust STS.
4. Click Token Creation under the WS-Trust STS tab.
5. Click Configure Token Creation.
6. Click IdP Token Processor Mapping on the Summary screen.

To modify an existing Token Processor Instance:

- Click its Name link.

To begin configuring an Token Processor Instance for this connection:

- Click Map New Token Processor Instance.

Selecting a Token Processor Instance

On this screen for a new connection, choose an instance of the Token Processor needed for this connection (see “Token Processors and Generators” on page 3).

You will use attributes returned from the token processor (the token-processor contract) to fulfill the attribute contract required for this partner and/or use them to look up additional attributes in a user data store. You make this choice on the next screen (see “Retrieving Attributes” on page 307).
Choose a Token Processor Instance from the drop-down list and click Next to continue.

To create or change a processor instance, as needed, click Manage Token Processor Instances.

To reach this screen for editing:
1. Click the connection name on the Main Menu. Click Manage All IdP, if needed, to see a full list of connections.
2. Click WS-Trust STS under the IdP Connection tab.
3. Click Configure WS-Trust STS.
4. Click Token Creation under the WS-Trust STS tab.
5. Click Configure Token Creation
6. Click IdP Token Processor Mapping on the Summary screen.
7. Click the Token Processor Instance Name.

Retrieving Attributes

For token creation, you can query local user data stores to help fulfill the attribute contract, in conjunction with attribute values supplied by the token processor you are using with PingFederate (see “Configuring Token Processors” on page 276).

The values supplied by the token processor are shown under Token Processor Contract on the Attribute Retrieval screen.
To reach this screen for editing:
1. Click the connection name on the Main Menu.
   Click Manage All IdP, if needed, to see a full list of connections.
2. Click WS-Trust STS under the IdP Connection tab.
3. Click Configure WS-Trust STS.
4. Click Token Creation under the WS-Trust STS tab.
5. Click Configure Token Creation
6. Click IdP Token Processor Mapping on the Summary screen.
7. Click the Token Processor Instance Name.
8. Click Attribute Retrieval on the Summary screen.

   ▶ If you choose to “Retrieve additional attributes . . .”, then you will identify a data store and specify lookup queries next (see the next section “Configuring Attribute Sources and User Lookup” on page 289).

   ▶ If you “Use only the Token Processor Contract values . . .”, then you will map values for the attribute contract next (see “Attribute Contract Fulfillment” on page 297).

**Tip:** To determine whether you need to look up additional values, compare the token-processor contract against the attribute contract (see “Defining an Attribute Contract” on page 285). If the attribute contract requires more information, determine whether a local data store can supply it. (You can also choose to use text constants or expressions for certain information—see “Attribute Contract Fulfillment” on page 297.)

### Configuring Attribute Sources and User Lookup

Attribute sources are specific database or directory locations containing information that may be needed for the attribute contract (see “Defining an Attribute Contract” on page 285). Attribute sources can be reused across connections to other SP partners.

This portion of the connection configuration allows you to set up search parameters for a data store.
Chapter 7
WS-Trust STS Configuration

To reach this screen for editing:

1. Click the connection name on the Main Menu.
   Click Manage All IdP, if needed, to see a full list of connections.
2. Click WS-Trust STS under the IdP Connection tab.
3. Click Configure WS-Trust STS.
4. Click Token Creation under the WS-Trust STS tab.
5. Click Configure Token Creation.
6. Click IdP Token Processor Mapping on the Summary screen.
7. Click the Token Processor Instance Name.
8. Click Attribute Source & User Lookup under the IdP Token Processor Mapping tab.
   If this step is not listed, then this instance is configured to use token-processor values only (see “Retrieving Attributes” on page 288).

To choose a Data Store:

▶ Choose an Active Data Store and click Next.

A data store configuration must be defined under System Settings for use within a connection. If the data store you want is not shown in the drop-down menu, click Manage Data Stores to add a new data store (see “Managing Data Stores” on page 72).

Attribute Source Setup

See the following sections in this manual, depending on the type of data store:

<table>
<thead>
<tr>
<th>Data Store Type</th>
<th>Related Manual Sections</th>
</tr>
</thead>
<tbody>
<tr>
<td>JDBC</td>
<td>“Selecting a JDBC Database Table and Columns” on page 291</td>
</tr>
<tr>
<td></td>
<td>“Configuring a Database Filter (WHERE Clause)” on page 292</td>
</tr>
<tr>
<td>LDAP</td>
<td>“Configuring an LDAP Directory Search” on page 294</td>
</tr>
<tr>
<td></td>
<td>“Configuring an LDAP Filter” on page 295</td>
</tr>
<tr>
<td>Custom</td>
<td>“Configuring Custom Source Filters” on page 297</td>
</tr>
<tr>
<td></td>
<td>“Selecting Custom Source Fields” on page 297</td>
</tr>
</tbody>
</table>
Selecting a JDBC Database Table and Columns

When you choose to use a database source for attributes, you follow this path through the configuration steps.

On this screen you begin to specify exactly where additional data can be found to complete the attribute contract when you send a security token to this SP (see “Defining an Attribute Contract” on page 285). Only one table may be used as a source of data for a JDBC lookup.

Field Descriptions

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schema</td>
<td>Lists the table structure that stores information within a database. Some databases, such as Oracle, require selection of a specific schema for a JDBC query. Other databases, such as MySQL, do not require selection of a schema.</td>
</tr>
<tr>
<td>Table</td>
<td>The name of the table contained in the database. Use the drop-down to change the table.</td>
</tr>
<tr>
<td>Columns to return from SELECT</td>
<td>Displays selected table columns. Select the columns that are associated with the desired attributes you would like to return from the JDBC query.</td>
</tr>
</tbody>
</table>

To reach this screen for editing:

1. Click the connection name on the Main Menu.
   - Click Manage All IdP, if needed, to see a full list of connections.
2. Click WS-Trust STS under the IdP Connection tab.
3. Click Configure WS-Trust STS.
4. Click Token Creation under the WS-Trust STS tab.
5. Click Configure Token Creation
6. Click IdP Token Processor Mapping on the Summary screen.
Chapter 7
WS-Trust STS Configuration

7. Click the Token Processor Instance Name.
8. Click Database Table and Columns under the IdP Token Processor Mapping tab.

To select a database table and columns for queries:
1. Choose a Schema file (when applicable) from the drop-down list.
2. Choose a Table from the drop-down list.
3. Choose a name under Columns to Return from Select and click Add Column.
   Repeat this step for other columns as needed.

**Note:** You do not need to add a column here for it to be used as part of a search key (see “Configuring a Database Filter (WHERE Clause)” next). Add only attributes from which you need actual values to pass in a token.

**Tip:** To determine what attributes to look up during a query, click the View Attribute Contract link to see what information must be collected (see “Defining an Attribute Contract” on page 285). Then determine what information is coming in from the token processor (see “Retrieving Attributes” on page 288). Information not contained in the token-processor contract may be pulled from the data store look-up query.

Configuring a Database Filter (WHERE Clause)

The JDBC WHERE clause in PingFederate queries the data table you selected to retrieve a record associated with a particular value (or values) from the incoming security token. The clause is in the form:

```
WHERE column1=value1 [AND column2=value2] [OR ...]
```

The left side of the first variable pair uses a column name in the database table you selected (see “Selecting a JDBC Database Table and Columns” on page 291).

The right side generally uses values passed in from a token processor (variables, including the correct formatting, are listed under Token Processor Values—see “Configuring Token Processors” on page 276).

You can also apply additional search criteria from your own database, using any other columns from the targeted table.

**Tip:** Click “View List of Columns . . .” to see a list from which to copy and paste.

For general information about WHERE clauses, consult your DBMS documentation.
EXAMPLE:

userid='\${username}'

In this example userid is the name of a column in the JDBC data store. On the right side, \${username} returns the value of the username variable from the IdP token processor.

**Important:** You must use the \$() syntax to retrieve the value of the enclosed variable and use single quotation marks around the \$() characters.

---

**Field Descriptions**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Where</td>
<td>WHERE clause statements conditionally select data from a table. Enter the WHERE clause statement in the space provided. For example: WHERE email='<a href="mailto:clive@company.com">clive@company.com</a>'.</td>
</tr>
</tbody>
</table>

---

**To reach this screen for editing:**

1. Click the connection name on the Main Menu.
   - Click Manage All IdP, if needed, to see a full list of connections.
2. Click WS-Trust STS under the IdP Connection tab.
3. Click Configure WS-Trust STS.
4. Click Token Creation under the WS-Trust STS tab.
5. Click Configure Token Creation
6. Click IdP Token Processor Mapping on the Summary screen.
7. Click the Token Processor Instance Name.
8. Click Database Filter under the IdP Token Processor Mapping tab.
**To construct the WHERE clause:**

1. Enter the statement in the space provided, following the guidelines and example above.
   
   The initial WHERE is optional.

2. Ensure the syntax and variable names are correct.

   When you click **Next**, you will map attribute values returned from the database into the security token (see “Attribute Contract Fulfillment” on page 297).

**Configuring an LDAP Directory Search**

When you choose to use an LDAP source for attributes, you follow this path through the configuration steps.

On this screen you specify the branch of your LDAP hierarchy where you want PingFederate to look up user data.

<table>
<thead>
<tr>
<th>Field Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Base DN</strong></td>
</tr>
<tr>
<td><strong>Search Scope</strong></td>
</tr>
<tr>
<td><strong>Root Object Class</strong></td>
</tr>
<tr>
<td><strong>Attributes to return from search</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base DN</td>
<td>The base distinguished name of the tree structure in which the search begins. This field is optional if records are located at the LDAP root.</td>
</tr>
<tr>
<td>Search Scope</td>
<td>Determines the node depth of the query. Select Subtree, One level or Object.</td>
</tr>
<tr>
<td>Root Object Class</td>
<td>The class containing the attributes you want.</td>
</tr>
<tr>
<td>Attributes to return from search</td>
<td>A list of added from the drop-down list below. Subject DN is a default attribute, which may be used as the primary user identifier.</td>
</tr>
</tbody>
</table>
To reach this screen for editing:
1. Click the connection name on the Main Menu.
   Click Manage All IdP, if needed, to see a full list of connections.
2. Click WS-Trust STS under the IdP Connection tab.
3. Click Configure WS-Trust STS.
4. Click Token Creation under the WS-Trust STS tab.
5. Click Configure Token Creation
6. Click IdP Token Processor Mapping on the Summary screen.
7. Click the Token Processor Instance Name.
8. Click LDAP Directory Search under the IdP Token Processor Mapping tab.

To select LDAP attributes:
1. (Optional) Enter a Base DN.
2. Select a Search Scope.
3. Select a Root Object Class.
4. Under Attributes to return from search, choose an attribute and click Add Attribute.
   Note that the attribute Subject DN is always returned by default.
5. Repeat the last step for other attributes as needed.
6. (Optional) Change the Search Scope or the Root Object Class if you want attributes from other locations.

Note: You do not need to add an attribute here for it to be used in a search filter (see “Configuring an LDAP Filter”). Add only attributes from which you need actual values to pass into the outgoing security token.

Configuring an LDAP Filter

The LDAP filter queries the data you selected to retrieve a record associated with a particular value (or values) from the incoming token. The filter is in the form:

```
attribute=$\{value\}
```

The left-side variable is an attribute you selected earlier (see “Configuring an LDAP Directory Search” on page 294).

The right side generally uses values passed in from the security token (variables, including the correct syntax, are listed under Security Token Values—see “Configuring Token Processors” on page 276).

You can also apply additional search criteria from your data store, using any other attributes from the targeted object classes.

Tip: Click “View List of Available LDAP Attributes” for a list from which you can copy and paste.

For general information about search filters, consult your LDAP documentation.
Field Descriptions

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filter</td>
<td>Narrows a search to locate requested data by either including or excluding specific records. An LDAP filter includes the attributes in the search and the value or range of values that the search is attempting to match. Searches are conducted by using three components: 1) at least one attribute (attribute data type) to search on, 2) a search filter operator that will determine what to match, and 3) the value of the attribute being sought. Searches must have at least one of each of these three components.</td>
</tr>
</tbody>
</table>

To reach this screen for editing:

1. Click the connection name on the Main Menu.
   - Click Manage All IdP, if needed, to see a full list of connections.
2. Click WS-Trust STS under the IdP Connection tab.
3. Click Configure WS-Trust STS.
4. Click Token Creation under the WS-Trust STS tab.
5. Click Configure Token Creation
6. Click IdP Token Processor Mapping on the Summary screen.
7. Click the Token Processor Instance Name.
8. Click LDAP Filter under the IdP Token Processor Mapping tab.
To construct the LDAP filter:

1. Enter the statement in the space provided, following the guidelines and example above.

   **Note:** If you used an anonymous binding to create this LDAP connection, your access might be restricted (see “Configuring an LDAP Connection” on page 77).

2. Ensure the syntax and variable names are correct.
3. Click Next.

**Configuring Custom Source Filters**

When you choose to use a custom source for attributes, you follow this path through the configuration steps.

On this screen you specify a filter, or lookup query, for your custom data source. This screen display and the syntax of the filter depends on your developer's implementation of the custom source SDK.

**Selecting Custom Source Fields**

On the Configure Custom Source Fields screen, you can choose from among the fields shown to map to the attribute contract. These choices are supplied by the driver implementation. Select only those needed to fulfill the attribute contract for this partner connection.

**Attribute Contract Fulfillment**

You map attributes for outgoing security tokens for this partner on the Attribute Contract Fulfillment screen.

<table>
<thead>
<tr>
<th>Attribute Contract</th>
<th>Source</th>
<th>Value</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAML_SUBJECT</td>
<td>LDAP</td>
<td>SubjectDN</td>
<td>None available</td>
</tr>
</tbody>
</table>

Map each attribute to fulfill the Attribute Contract from one of these Sources:

- **Token**
  
  When you make this selection, the associated Value drop-down list is populated by the token processor.

- **LDAP/JDBC/Custom**
  
  Values are returned from your attribute source (if you are using data store—see “Retrieving Attributes” on page 288). When you make this selection, the Value list is populated by the LDAP, JDBC or Custom attributes you identified as an Attribute Source (see “Configuring an LDAP Directory Search” on page 294,
“Selecting a JDBC Database Table and Columns” on page 291, or “Configuring Custom Source Filters” on page 297).

- **Request**
  Values are supplied from parameters in the token request received from the Web Service Client. This selection is available only if a Request Contract was selected earlier (see “Selecting a Request Contract” on page 286).

- **Expression (when enabled)**
  This option provides more complex mapping capabilities—for example, transforming incoming values into different formats (see “Using Attribute Mapping Expressions” on page 353). All of the variables available for text entries (see below) are also available for expressions.

- **Text**
  The value is what you enter. This can be text only, or you can mix text with references to any of the values from the incoming token, using the `attribute` syntax.
  You can also enter values from your data store, when applicable, using this syntax:
  
  ```
  ${ds.attribute}
  ```
  
  where `attribute` is any of the data store attributes you have selected.

**To reach this screen for editing:**

1. Click the connection name on the Main Menu.
   - Click Manage All IdP, if needed, to see a full list of connections.
2. Click **WS-Trust STS** under the IdP Connection tab.
3. Click **Configure WS-Trust STS**.
4. Click **Token Creation** under the WS-Trust STS tab.
5. Click **Configure Token Creation**
6. Click **IdP Token Processor Mapping** on the Summary screen.
7. Click the Token Processor Instance Name.
8. Click **Attribute Contract Fulfillment** under the IdP Token Processor Mapping tab.

**To map attributes:**

1. Choose a Source for each Target attribute.
2. Choose (or enter) a Value for each Attribute.
   - See “Map each attribute to fulfill the Attribute Contract from one of these Sources:” on page 297. All values must be mapped.
3. Click **Next**.

**Using the Mapping Summary Screen**

When you have finished configuring IdP Token Processor Mapping, you can review the configuration on the Summary screen. If you need to make any changes, click the heading over the information you want to edit.

- If you are editing an existing connection, click **Done** on successive screens until you reach the WS-Trust STS screen, and then click **Save**.

**To save a new configuration:**

1. Click **Done** to return to the IdP Token Processor Mapping screen.
2. Click **Next** to go to the Token Creation Summary screen, and then click **Done**.
3. On the Token Creation screen, click **Done**.
4. On the WS-Trust STS screen, click **Save**.

**Request Error Handling**

If you are using request parameters to fulfill the attribute contract and the parameter values are not supplied, you can choose whether to continue or abort the token-creation process.

### Using the Token Creation Summary Screen

When you have finished configuring Token Creation, you can review the configuration on the Summary screen. If you need to make any changes, click the heading over the information you want to edit.

- If you are editing an existing connection, click **Done** on successive screens until you reach the WS-Trust STS screen, and then click **Save**.
Chapter 7
WS-Trust STS Configuration

SP Configuration for STS

This section covers the SP configuration for STS, which involves:

- “Configuring Token Generators”
- “Configuring IdP Connections for STS”

Configuring Token Generators

Token Generators are used to issue security tokens that can be consumed by Web Services at your site (see “Token Processors and Generators” on page 3). Token Generators for SAML 2.0 and SAML 1.1 tokens are included with the PingFederate installation. This section provides guidance on configuring “instances” of either of the SAML Token Generators. You must configure at least one generator in order to set up an STS connection.

Only one instance of any Token Generator may be configured.

Additional Token Generators are available from the Ping Identity Web site (at www.pingidentity.com/products/PingFederate-Token-Translators.cfm).

To begin configuring SAML 1.1 or 2.0 Token Generators:

- On the Main Menu, click Token Generators under Application Integration Settings for My SP Configuration.

  If this link is not shown, ensure that the WS-Trust STS is enabled in Server Settings (see “Enabling the WS-Trust STS” on page 269).

To configure a new token-generator instance:

- Click Create New Instance

To edit an existing instance:

- Click the Instance Name and click the step you need to change.

To delete an instance:

1. Click Delete next to the Instance Name. (To undo the deletion, click Undelete.)

   Note: This option is available only if the generator instance is not in use for a connection.

2. Click Save to confirm the deletion.
Selecting a Token Generator Type

The first step in creating a SAML token-generator instance is choosing the generator type.

To define an instance:
1. Enter the Instance Name and Instance Id on the Type screen.
2. Select SAML 1.1 Token Generator <version> or SAML 2.0 Token Generator <version> from the drop-down menu.
3. Click Next.

Configuring a Token Generator Instance

On the Instance Configuration screen, you specify parameters for generated SAML tokens.
Chapter 7
WS-Trust STS Configuration

To configure the token-generator instance:

- Provide information for all required fields—refer to screen Descriptions for information about each field.

  If you have not yet imported a Signing Certificate, click Manage Signing Certificates.

  The Audience entry is used for the <audience> element of the generated SAML token.

Extending a Generator Contract

Token generators allow administrators to add to a built-in list of user attributes that the generator includes in the outgoing token—an extended generator-attribute contract.
To add an attribute:

- Enter the attribute name in the text box and click Add.

**Editing and Saving Generator Instances**

From the Summary screen, you can reach token-generator settings for editing.

**To edit the configuration:**

1. Click the heading above the information you want to change.
2. Make your changes.
3. Click **Done** on the configuration page and **Save** on the Manage Token Generators screen.

**To save a generator instance:**

1. Click **Done** on the Summary screen.
2. Click **Save** on the Manage Token Generators screen.

**Configuring IdP Connections for STS**

You can configure an STS connection to an IdP partner either in conjunction with browser-based SSO or independently.

**To enable STS for a new connection, or to add the capability to an existing connection:**

- Select the WS-Trust STS option on the Connection Type screen (see “Choosing a Connection Type” on page 210).

**Note:** Before this option can be selected, the WS-Trust protocol must be enabled in Server Settings (see “Server Settings” on page 269).

When the option is enabled, the configuration starts on the WS-Trust STS screen.
To continue, click **Configure WS-Trust STS**.

### Configuring Protocol Settings

On the Protocol Settings screen, choose whether to validate incoming SAML tokens or to validate and then also generate different tokens to enable SSO access to Web Services at your site.

Also on this screen, if incoming SAML 2.0 tokens for this connection are required to be encrypted, select the checkbox for decrypting assertions. When you make this selection, you will be required to choose a decryption certificate for this partner later in the connection configuration (if one is not already selected for Browser SSO purposes—see “Selecting a Decryption Key” on page 264).

If you choose not to generate new tokens, then no further settings are needed for this task—click **Next** and refer to “Using the Token Generation Summary Screen” on page 318 for instructions on saving this configuration.

You will be asked later to choose a certificate with which to verify the signature on the incoming SAML token (see “Configuring Signature Verification Settings” on page 165).

### Configuring Token Generation

For the PingFederate STS to issue a security token that meets identity requirements of Web Services at your site, you must indicate what user attributes are included in the incoming token (the “attribute contract”). The attribute values from the incoming token can be then mapped to attributes in the token generator you select (see “Attribute Contract Fulfillment” on page 316). As with Browser SSO,
mapping can be augmented using local data stores, variable or constant text, or expressions.

Details of this configuration are handled under the Token Generation task.

To continue, click **Configure Token Creation**.

### Defining an Attribute Contract

An attribute contract is the set of user attributes expected in incoming security tokens (see “Attribute Contracts” on page 7). You identify these attributes on this screen.

To reach this screen for editing:

1. Click the connection name on the Main Menu.
   
   Click **Manage All SP**, if needed, to see a full list of connections.

2. Click **WS-Trust STS** under the SP Connection tab.

3. Click **Configure WS-Trust STS**.

4. Click **Token Generation** under the WS-Trust STS tab.
   
   If this step is not shown, token generation is not selected for the connection (see “Configuring Protocol Settings” on page 304).

5. Click **Configure Token Generation**
6. Click Attribute Contract on the Summary screen.

**To add an attribute:**
1. Enter the attribute name in the text box.
   Attribute names are case-sensitive and must correspond to the names expected by the requester.
2. (Optional) Select the checkbox under Mask Values in Log.
3. Click Add.

**To modify an attribute name:**
1. Click Edit under Action for the attribute.
2. Make the change and click Update.

**To delete an attribute:**
- Click Delete under Action for the attribute.

---

**Mapping Token Generators**

Token generators provide a mechanism through which PingFederate can generate a local token based upon an incoming SAML token, including mapping user attributes to be included in the generated token. A configured and deployed token generator in PingFederate is known as a token-generator instance.

You can map one or more token generator instances into each IdP connection to satisfy multiple session-management requirements where needed. The same instances may be mapped by multiple connections.

The configuration begins on the Token Generator Mapping & User Lookup screen. If you have not yet configured an instance of a token generator you need for this connection, see “Configuring Token Generators” on page 300.

---

**To reach this screen for editing:**
1. Click the connection name on the Main Menu.
2. Click Manage All SP, if needed, to see a full list of connections.
3. Click WS-Trust STS under the SP Connection tab.
4. Click Configure WS-Trust STS.
5. Click Token Generation under the WS-Trust STS tab.
   - If this step is not shown, token generation is not selected for the connection (see “Configuring Protocol Settings” on page 304).
6. Click Configure Token Generation
6. Click Token Generator Mapping & User Lookup on the Summary screen.

To modify an existing Token Generator Instance:

- Click its Name link.

To begin configuring an Token Generator Instance for this connection:

- Click Map New Token Generator Instance.

Selecting a Token Generator Instance

On this screen for a new connection, choose an instance of the Token Generator needed for this connection (see “Token Processors and Generators” on page 3).

You will use attributes contained in the incoming security token to fulfill the token generator contract for this STS connection and/or use them to look up additional attributes in a user data store. You make this choice on the next screen (see “Retrieving Attributes” on page 288).

Choose a Token Generator Instance from the drop-down list and click Next to continue.

To create or change a Token Generator Instance, as needed, click Manage Token Generator Instances.

Retrieving Attributes

For token generation, you can query local user data stores to help fulfill the token-generator contract, in conjunction with attribute values supplied by the incoming token.

The values supplied by the token are shown under Attribute Contract on the Attribute Retrieval screen.
To reach this screen for editing:

1. Click the connection name on the Main Menu. Click Manage All SP, if needed, to see a full list of connections.
2. Click WS-Trust STS under the SP Connection tab.
3. Click Configure WS-Trust STS.
4. Click Token Generation under the WS-Trust STS tab.
   If this step is not shown, token generation is not selected for the connection (see “Configuring Protocol Settings” on page 304).
5. Click Configure Token Generation
6. Click Token Generator Mapping & User Lookup on the Summary screen.
7. Click the Token Generator Instance Name.
8. Click Attribute Retrieval on the Summary screen.
   ▶ If you choose to look up additional information, then you will identify a data store and specify lookup queries next (see the next section “Choosing a Data Store” on page 308).
   ▶ If you use only the attributes available (the default), then you will map values for the attribute contract next (see “Attribute Contract Fulfillment” on page 316).

Tip: To determine whether you need to look up additional values, compare the attribute contract against the token-generator contract on the previous screen (see “Selecting a Token Generator Instance” on page 307). If the token-generator contract requires more information, determine whether your local data stores can supply it. (You can also choose to use text constants or expressions for certain information—see “Attribute Contract Fulfillment” on page 316.)

Choosing a Data Store

This portion of the connection configuration allows you to set up search parameters for a data store.
To reach this screen for editing:
1. Click the connection name on the Main Menu.
   Click Manage All SP, if needed, to see a full list of connections.
2. Click WS-Trust STS under the SP Connection tab.
3. Click Configure WS-Trust STS.
4. Click Token Generation under the WS-Trust STS tab.
   If this step is not shown, token generation is not selected for the connection (see “Configuring Protocol Settings” on page 304).
5. Click Configure Token Generation
6. Click Token Generator Mapping & User Lookup on the Summary screen.
7. Click the Token Generator Instance Name.
8. Click Data Store under the Token Generator Mapping & User Lookup tab.
   If this step is not presented, this Token Generator Instance is not configured to look up user attributes in a data store (see “Retrieving Attributes” on page 307).

To choose a Data Store:

- Choose an Active Data Store and click Next.
  A data store configuration must be defined under System Settings for use within a connection. If the data store you want is not shown in the drop-down menu, click Manage Data Stores to add a new data store (see “Managing Data Stores” on page 72).

Attribute Source Setup

See the following sections in this manual, depending on the type of data store:

<table>
<thead>
<tr>
<th>Data Store Type</th>
<th>Related Manual Sections</th>
</tr>
</thead>
<tbody>
<tr>
<td>JDBC</td>
<td>“Selecting a JDBC Database Table and Columns” on page 310</td>
</tr>
<tr>
<td></td>
<td>“Configuring a Database Filter (WHERE Clause)” on page 311</td>
</tr>
<tr>
<td>LDAP</td>
<td>“Configuring an LDAP Directory Search” on page 313</td>
</tr>
<tr>
<td></td>
<td>“Configuring an LDAP Filter” on page 314</td>
</tr>
<tr>
<td>Custom</td>
<td>“Configuring Custom Source Filters” on page 316</td>
</tr>
<tr>
<td></td>
<td>“Selecting Custom Source Fields” on page 316</td>
</tr>
</tbody>
</table>
**Selecting a JDBC Database Table and Columns**

When you choose to use a database source for attributes, you follow this path through the configuration steps.

On this screen you begin to specify exactly where additional data can be found to complete the token-generator contract (see “Retrieving Attributes” on page 307). Only one table may be used as a source of data for a JDBC lookup.

**Field Descriptions**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schema</td>
<td>Lists the table structure that stores information within a database. Some databases, such as Oracle, require selection of a specific schema for a JDBC query. Other databases, such as MySQL, do not require selection of a schema.</td>
</tr>
<tr>
<td>Table</td>
<td>The name of the table contained in the database. Use the drop-down to change the table.</td>
</tr>
<tr>
<td>Columns to return from SELECT</td>
<td>Displays selected table columns. Select the columns that are associated with the desired attributes you would like to return from the JDBC query.</td>
</tr>
</tbody>
</table>

**To reach this screen for editing:**

1. Click the connection name on the Main Menu.
   
   Click **Manage All SP**, if needed, to see a full list of connections.
2. Click **WS-Trust STS** under the SP Connection tab.
3. Click **Configure WS-Trust STS**.
4. Click **Token Generation** under the WS-Trust STS tab.
   
   If this step is not shown, token generation is not selected for the connection (see “Configuring Protocol Settings” on page 304).
5. Click **Configure Token Generation**
6. Click **Token Generator Mapping & User Lookup** on the Summary screen.
7. Click the Token Generator Instance Name.
8. Click **Database Table and Columns** under the Token Generator Mapping & User Lookup tab.

**To select a database table and columns for queries:**

1. Choose a Schema file (when applicable) from the drop-down list.
2. Choose a Table from the drop-down list.
3. Choose a name under Columns to Return from Select and click **Add Column**. Repeat this step for other columns as needed.

**Note:** You do not need to add a column here for it to be used as part of a search key (see “Configuring a Database Filter (WHERE Clause)” next). Add only attributes from which you need actual values to pass in a token.

**Tip:** To determine what attributes to look up during a query, click the **View Attribute Contract** link to see what information must be collected (see “Defining an Attribute Contract” on page 305). Then determine what information is coming in from the token processor (see “Retrieving Attributes” on page 307). Information not contained in the token-processor contract may be pulled from the data store look-up query.

**Configuring a Database Filter (WHERE Clause)**

The JDBC **WHERE** clause in PingFederate queries the data table you selected to retrieve a record associated with a particular value (or values) from the incoming security token. The clause is in the form:

```
WHERE column1=value1 [AND column2=value2] [OR ...]
```

The left side of the first variable pair uses a column name in the database table you selected (see “Selecting a JDBC Database Table and Columns” on page 310).

The right side generally uses values passed in from the incoming SAML token (variables, including the correct formatting, are listed under **Assertion Values**).

You can also apply additional search criteria from your own database, using any other columns from the targeted table.

**Tip:** Click **“View List of Columns . . .”** to see a list from which to copy and paste.

For general information about **WHERE** clauses, consult your DBMS documentation.

**EXAMPLE:**

```plaintext
userid='${username}'
```

In this example `userid` is the name of a column in the JDBC data store. On the right side, `'$\{username\}'` returns the value of the `username` variable from the IdP token processor.

**Important:** You must use the `$()` syntax to retrieve the value of the enclosed variable and use single quotation marks around the `$()` characters.
### Field Descriptions

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Where</td>
<td>WHERE clause statements conditionally select data from a table. Enter the WHERE clause statement in the space provided. For example: WHERE email='<a href="mailto:clive@company.com">clive@company.com</a>'.</td>
</tr>
</tbody>
</table>

**To reach this screen for editing:**

1. Click the connection name on the Main Menu.
   
   Click Manage All SP, if needed, to see a full list of connections.

2. Click WS-Trust STS under the SP Connection tab.

3. Click Configure WS-Trust STS.

4. Click Token Generation under the WS-Trust STS tab.
   
   If this step is not shown, token generation is not selected for the connection (see “Configuring Protocol Settings” on page 304).

5. Click Configure Token Generation

6. Click Token Generator Mapping & User Lookup on the Summary screen.

7. Click the Token Generator Instance Name.

8. Click Database Filter from the steps list under the Token Generator Mapping & User Lookup tab.

**To construct the WHERE clause:**

1. Enter the statement in the space provided, following the guidelines and example above.
   
   The initial WHERE is optional.

2. Ensure the syntax and variable names are correct.
   
   When you click Next, you will map attribute values returned from the database into the security token (see “Attribute Contract Fulfillment” on page 316).
Configuring an LDAP Directory Search

When you choose to use an LDAP source for attributes, you follow this path through the configuration steps.

On this screen you specify the branch of your LDAP hierarchy where you want PingFederate to look up user data.

Field Descriptions

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base DN</td>
<td>The base distinguished name of the tree structure in which the search begins. This field is optional if records are located at the LDAP root.</td>
</tr>
<tr>
<td>Search Scope</td>
<td>Determines the node depth of the query. Select Subtree, One level or Object.</td>
</tr>
<tr>
<td>Root Object Class</td>
<td>The class containing the attributes you want.</td>
</tr>
<tr>
<td>Attributes to return from search</td>
<td>A list of added from the drop-down list below. Subject DN is a default attribute, which may be used as the primary user identifier.</td>
</tr>
</tbody>
</table>

To reach this screen for editing:

1. Click the connection name on the Main Menu.
   Click Manage All SP, if needed, to see a full list of connections.
2. Click WS-Trust STS under the SP Connection tab.
3. Click Configure WS-Trust STS.
4. Click Token Generation under the WS-Trust STS tab.
   If this step is not shown, token generation is not selected for the connection (see “Configuring Protocol Settings” on page 304).
5. Click Configure Token Generation
6. Click Token Generator Mapping & User Lookup on the Summary screen.
7. Click the Token Generator Instance Name.
8. Click **LDAP Directory Search** under the Token Generator Mapping & User Lookup tab.

**To select LDAP attributes:**
1. (Optional) Enter a Base DN.
2. Select a Search Scope.
3. Select a Root Object Class.
4. Under Attributes to return from search, choose an attribute and click Add Attribute.
   Note that the attribute Subject DN is always returned by default.
5. Repeat the last step for other attributes as needed.
6. (Optional) Change the Search Scope or the Root Object Class if you want attributes from other locations.

![Note: You do not need to add an attribute here for it to be used in a search filter (see “Configuring an LDAP Filter”). Add only attributes from which you need actual values to pass into the outgoing security token.]

**Configuring an LDAP Filter**

The LDAP filter queries the data you selected to retrieve a record associated with a particular value (or values) from the incoming token. The filter is in the form:

```
attribute=${value}
```

The left-side variable is an attribute you selected earlier (see “Configuring an LDAP Directory Search” on page 313).

The right side generally uses values passed in from the incoming SAML token (variables, including the correct syntax, are listed under Assertion Values).

You can also apply additional search criteria from your data store, using any other attributes from the targeted object classes.

![Tip: Click “View List of Available LDAP Attributes” for a list from which you can copy and paste.]

For general information about search filters, consult your LDAP documentation.
Field Descriptions

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filter</td>
<td>Narrows a search to locate requested data by either including or excluding specific records. An LDAP filter includes the attributes in the search and the value or range of values that the search is attempting to match. Searches are conducted by using three components: 1) at least one attribute (attribute data type) to search on, 2) a search filter operator that will determine what to match, and 3) the value of the attribute being sought. Searches must have at least one of each of these three components.</td>
</tr>
</tbody>
</table>

**To reach this screen for editing:**

1. Click the connection name on the Main Menu.
   - Click Manage All SP, if needed, to see a full list of connections.
2. Click WS-Trust STS under the SP Connection tab.
3. Click Configure WS-Trust STS.
4. Click Token Generation under the WS-Trust STS tab.
   - If this step is not shown, token generation is not selected for the connection (see “Configuring Protocol Settings” on page 304).
5. Click Configure Token Generation
6. Click Token Generator Mapping & User Lookup on the Summary screen.
7. Click the Token Generator Instance Name.
8. Click LDAP Filter under the Token Generator Mapping & User Lookup tab.
To construct the LDAP filter:

1. Enter the statement in the space provided, following the guidelines and example above.

   **Note:** If you used an anonymous binding to create this LDAP connection, your access might be restricted (see “Configuring an LDAP Connection” on page 77).

2. Ensure the syntax and variable names are correct.
3. Click Next.

**Configuring Custom Source Filters**

When you choose to use a custom source for attributes, you follow this path through the configuration steps.

On this screen you specify a filter, or lookup query, for your custom data source. This screen display and the syntax of the filter depends on your developer's implementation of the custom source SDK.

**Selecting Custom Source Fields**

On the Configure Custom Source Fields screen, you can choose from among the fields shown to map to the token processor contract. These choices are supplied by the driver implementation. Select only those needed to fulfill the attribute contract for this partner connection.

**Attribute Contract Fulfillment**

You map attributes for outgoing security tokens for this partner on the Token Generator Contract Fulfillment screen.

Map each attribute to fulfill the Token Generator Contract from one of these Sources:

- **Assertion**
  
  When you make this selection, the associated Value drop-down list is populated by the incoming SAML token (“Assertion”).

- **LDAP/JDBC/Custom**
  
  Values are returned from the selected data store (see “Retrieving Attributes” on page 307). When you make this selection, the Value list is populated by the LDAP, JDBC or Custom attributes specified in previous screens (see “Configuring an LDAP Directory Search” on page 313, “Selecting a JDBC Database Table and Columns” on page 310, or “Configuring Custom Source Filters” on page 316).
Expression (when enabled)
This option provides more complex mapping capabilities—for example, transforming incoming values into different formats (see “Using Attribute Mapping Expressions” on page 353). All of the variables available for text entries (see below) are also available for expressions.

Text
The value is what you enter. This can be text only, or you can mix text with references to any of the values from the incoming token, using the ${attribute} syntax.
You can also enter values from your data store, when applicable, using this syntax:
${ds.attribute}
where attribute is any of the data store attributes you have selected.

To reach this screen for editing:
1. Click the connection name on the Main Menu.
   Click Manage All SP, if needed, to see a full list of connections.
2. Click WS-Trust STS under the SP Connection tab.
3. Click Configure WS-Trust STS.
4. Click Token Generation under the WS-Trust STS tab.
   If this step is not shown, token generation is not selected for the connection (see “Configuring Protocol Settings” on page 304).
5. Click Configure Token Generation
6. Click Token Generator Mapping & User Lookup on the Summary screen.
7. Click the Token Generator Instance Name.
8. Click Token Generator Contract Fulfillment under the Token Generator Mapping & User Lookup tab.

To map attributes:
1. Choose a Source for each Target attribute.
2. Choose (or enter) a Value for each Attribute.
   See “Map each attribute to fulfill the Token Generator Contract from one of these Sources:” on page 316. All values must be mapped.
3. Click Next.

Using the Mapping Summary Screen
When you have finished configuring Token Generator Mapping & User Lookup, you can review the configuration on the Summary screen. If you need to make any changes, click the heading over the information you want to edit.

If you are editing an existing connection, click Done on successive screens until you reach the WS-Trust STS screen, and then click Save.

To save a new configuration:
1. Click Done to return to the Token Generator Mapping & User Lookup screen.
2. Click Next to go to the Token Generation Summary screen, and then click Done.
3. On the Token Generation screen, click Done.
4. On the WS-Trust STS screen, click Save.
Using the Token Generation Summary Screen

When you have finished configuring Token Generation, you can review the configuration on the Summary screen. If you need to make any changes, click the heading over the information you want to edit.

► If you are editing an existing connection, click **Done** on successive screens until you reach the WS-Trust STS screen, and then click **Save**.

On the WS-Trust STS Summary screen, you can review the configuration for this connection.

► If you need to make any changes to a new or existing connection, click the heading over the information you want to modify.
OpenToken Adapter Configuration

In order to transfer identity and other user information between the PingFederate server and an end application, the product architecture allows for custom adapters to be deployed with the server (see “SSO Integration Kits and Adapters” on page 4).

PingFederate ships with a deployed OpenToken Adapter, which uses a secure token format (OpenToken) to transfer user attributes between an application and the PingFederate server. On the IdP side, the OpenToken Adapter allows the PingFederate server to receive a user's identity from the IdP application. On the SP side, the OpenToken Adapter can be used to transfer user-identity information to the target SP application.

Specialized application integration kits are available from www.pingidentity.com. Many kits leverage the OpenToken Adapter to integrate applications with the PingFederate server. The agent portions of the integration kits reside with the application and use the OpenToken to communicate with the OpenToken Adapter.

Note: To integrate applications for use with the OpenToken Adapter, download an integration kit for PingFederate from www.pingidentity.com and follow instructions for installing and using Agent Toolkits in the accompanying documentation. Follow the configuration instructions in this appendix to set up the OpenToken Adapter to use with your applications.
The following figure shows a basic IdP-initiated SSO scenario using PingFederate with the Java Integration Kit on both sides of an identity federation.

![Diagram of SSO scenario](image)

**Figure 4:** SP-Initiated SSO: POST/POST

**Processing Steps:**

1. A user initiates an SSO transaction.
2. The IdP application inserts attributes into the Agent Toolkit for Java, which encrypts the data internally and generates an OpenToken.
   Attributes are encrypted and decrypted using the Java Cryptography Extension (JCE). For more information, see [http://java.sun.com/products/jce](http://java.sun.com/products/jce).
3. A request containing the OpenToken is redirected to the PingFederate IdP server.
4. The server invokes the OpenToken IdP Adapter, which retrieves the OpenToken, decrypts, parses, and passes it to the PingFederate IdP server. The PingFederate IdP server then generates a Security Assertion Markup Language (SAML) assertion.
5. The SAML assertion is sent to the SP site.
6. The PingFederate SP server parses the SAML assertion and passes the user attributes to the OpenToken SP Adapter. The Adapter encrypts the data internally and generates an OpenToken.
7. A request containing the OpenToken is redirected to the SP application.
8. The Agent Toolkit for Java decrypts and parses the OpenToken and makes the attributes available to the SP Application.

**Configuring the IdP OpenToken Adapter**

1. If you have not already done so, log on to the PingFederate administrative console and click **Adapters** under My IdP Configuration on the Main Menu.
2. On the Manage Adapter Instances screen, click **Create New Adapter Instance**.
3. On the adapter Type screen, enter an Instance Name and Instance Id, select OpenToken Adapter 2.3 (or higher) as the Type, and click Next.

   The Instance Id may not contain spaces or underscores.

4. On the IdP Adapter screen, enter the values as described for the adapter configuration.

   These values are dependent on your developer's implementation.

5. (Optional) Click Show Advanced Fields to reconfigure default settings for the OpenToken, as needed.

   Refer to the on-screen field descriptions for more information.

6. Click Next.
7. On the Actions screen, click **Download** under Action Invocation Link.

8. On the next screen, click **Export** and save the properties file.

   The values in the resulting file, `agent-config.txt`, represents the console configuration and are used by the IDP application. Refer to your respective Integration Kit User Guide for more information.

9. (Optional) On the Extended Contract screen, you can configure additional attributes for the adapter (see “Extending an Adapter Contract” on page 105).

10. Click **Next**.

11. On the Adapter Attributes screen, select the subject checkbox under Pseudonym (optionally, select other attributes, if you added any at Step 9).

   This selection is used if any of your SP partners will make use of pseudonyms for account linking (see “Account Linking” on page 5).
You can also choose to mask the values of any or all attributes that PingFederate logs from the adapter at runtime (see “Attribute Masking” on page 9).

12. Click Next.
13. On the Summary screen, review the configuration and click Done.
   You can also click any heading to go back and change information.
14. On the Manage Adapter Instances screen, click Save.

**Important:** You must click Save if you wish to retain the adapter configuration.

### Configuring the SP OpenToken Adapter

1. If you have not already done so, log on to the PingFederate administrative console and click Adapters under My SP Configuration on the Main Menu.
2. On the Manage Adapter Instances screen, click Create New Adapter Instance.
3. Enter an Instance Name and Instance Id, select OpenToken Adapter 2.3 (or higher) as the Type, and click Next.
   The Instance Id may not contain spaces or underscores.

4. Enter values for the adapter configuration on the Instance Configuration screen.
   These values are dependent on your developer's implementation.
5. (Optional) Click **Show Advanced Fields** to reconfigure default settings for the OpenToken, as needed. 
   Refer to the on-screen descriptions for more information.

6. Click **Next**.

7. On the Actions screen, click **Download** under Action Invocation Link.

8. On the next screen, click **Export** and save the properties file.
   The values in the resulting file, agent-config.txt, are set by the console configuration and used by the SP application. Refer to your respective Integration Kit User Guide for more information.

9. Click **Next**.

10. (Optional) On the Extended Contract screen, you can configure additional attributes for the adapter (see “Extending an Adapter Contract” on page 105).

11. Click **Next**.

12. On the Summary screen, review the configuration and click **Done**.
   You can also click any heading to go back and change information.

13. On the Manage Adapter Instances screen, click **Save**.

   **Important:** You must click **Save** if you wish to retain the adapter configuration.
**Note:** If this is the second instance of an OpenToken Adapter configuration, then you must first click **Next** and map target URLs to adapter instances (see “Mapping URLs to Adapter Instances” on page 198).
Appendix A

OpenToken Adapter Configuration
LDAP Adapter Configuration

Initial user authentication is normally handled outside of the PingFederate server using an application or IdM system logon module. PingFederate’s adapter and application agents are typically used to integrate with these local authentication mechanisms (see “SSO Integration Kits and Adapters” on page 4).

PingFederate packages an LDAP Authentication Service Adapter (“LDAP Adapter”) and logon form that can authenticate users directly against an LDAP data store. This adapter may be used if your organization does not have a centralized local authentication service and your user stores are maintained by LDAP servers.

On the IdP side, when the PingFederate IdP server receives an authentication request for SP-initiated SSO or the user clicks a link for IdP-initiated SSO, the IdP server invokes the LDAP Adapter and prompts the user for local IdP credentials. The credentials are then compared against the LDAP server and, if validated, a SAML assertion is generated.

On the SP side, local user logon is needed only for account linking. In this federation scenario, the IdP generates a name identifier (which may be a pseudonym) that must be associated with a local user ID used at the SP (see “Account Linking” on page 5).

PingFederate and the LDAP SP adapter handle account linking in the following way:

1. The adapter prompts the user for local SP credentials and validates the credentials against the data store.

2. The adapter passes the user ID to PingFederate to save in an embedded account-linking data store. (You can use the SDK to extend account linking to an external data store.)
3. The LDAP SP adapter then uses the PingFederate Java, .NET, or PHP Integration Kit to transfer user attributes to the SP application (see “OpenToken Adapter Configuration” on page 319).

**Note:** Application-integration kits for Java, .NET, and PHP are available separately from Ping Identity at [www.pingidentity.com](http://www.pingidentity.com). To integrate SP applications for the LDAP Adapter, unzip the respective integration kit and follow instructions for installing and using the Agent Toolkits in the accompanying documentation. Then follow the configuration instructions in this appendix to set up the LDAP Adapter to use with your application(s).

The SP application integration consists of two parts. The first is configuration of the adapter, which runs on the PingFederate server and is the subject of this appendix. The second part is an Agent Toolkit, which resides with the application server.

The following figure shows a basic SP-initiated SSO scenario with PingFederate servers using the LDAP Adapter, which is integrated on both sides of the identity federation.

### Processing Steps

1. The user initiates an SSO transaction from an external SP application.
2. The external SP application starts the SSO process through the federation SP server.
3. The request is sent to the federation IdP server.
4. The LDAP IdP adapter authenticates the user against an LDAP server and passes the authentication to the federation IdP server.
5. The federation IdP server generates a SAML assertion and the request is redirected to the SP site.
6. The federation SP server parses the SAML assertion and passes the user attributes to the LDAP SP adapter. The LDAP SP adapter authenticates the user against an LDAP server using account linking, encrypts the data internally, and generates an OpenToken.
7. A request containing an OpenToken is redirected to the SP application.
8. The SP Agent Toolkit decrypts and parses the OpenToken and makes the attributes available to the SP application.

Configuring the IdP LDAP Adapter

1. If you have not already done so, establish a connection between PingFederate and your LDAP server (see “Configuring an LDAP Connection” on page 77).
2. Click **IdP Adapters** on the Main Menu screen.
3. On the Manage Adapter Instances screen, click **Create New Adapter Instance**.
4. On the adapter Type screen, enter an Instance Name and Instance Id, select LDAP Authentication Service 2.0 as the Type, and click **Next**. The Instance Id may not contain spaces or underscores.

5. On the IdP Adapter screen, enter the values for adapter configuration described below.

   ![IdP Adapter Configuration Screen]

   **Note:** If you do not know the values to enter at this time, select your LDAP server and enter placeholders (in any format) for the rest of the entry fields. You can return to this screen to enter the correct values at any time (see “Configuring IdP Adapters” on page 102). Click **Manage Data Stores** if you have not established a connection to your LDAP server.
### Appendix B

**LDAP Adapter Configuration**

#### Property Description

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDAP Datastore</td>
<td>The LDAP Data store configured in PingFederate.</td>
</tr>
<tr>
<td>Search Base</td>
<td>The location in the LDAP directory server from which the search begins.</td>
</tr>
<tr>
<td>Search Filter</td>
<td>Query used to produce the desired set of matching records.</td>
</tr>
<tr>
<td>Realm</td>
<td>The name of a protected area. The value of this field is sent as a part of the HTTP basic authentication request. It appears in the dialog box that prompts the user for a username and password.</td>
</tr>
</tbody>
</table>
6. (Optional) Click **Show Advanced Fields** and change parameters as needed.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scope of Search</td>
<td>The level of search to be performed in the search base. One level indicates a search of objects immediately subordinate to the base object, not including the base object itself. Subtree indicates a search of the base object and the entire subtree within the base object distinguished name.</td>
</tr>
<tr>
<td>Connection Pooling</td>
<td>A type of connection sharing supported by the LDAP server, which maintains a pool of (possibly) previously used connections and assigns them as needed.</td>
</tr>
<tr>
<td>Operational Mode</td>
<td>The method of interaction between the adapter and user agent. In HTTP Basic Authentication mode, the adapter interacts with the user via HTTP basic authentication. In HTML Form Authentication mode, the adapter uses an HTML form.</td>
</tr>
<tr>
<td>Challenge Retries</td>
<td>The number of attempts allowed for password authentication.</td>
</tr>
</tbody>
</table>

7. Click **Next**.

8. On the Adapter Attributes screen, select the subject checkbox under **Pseudonym** (and, optionally, other attributes, if available).

This selection is used if any of your SP partners will make use of pseudonyms for account linking (see “Account Linking” on page 5).
Note: A selection is required regardless of whether you will use pseudonyms for account linking. This allows account linking to be used later without having to delete and reconfigure the adapter. Ensure that you choose at least one attribute that is unique for each user (for example, email) to prevent the same pseudonym from being assigned to multiple users.

You can also choose to mask the values of any or all attributes that PingFederate logs from the adapter at runtime (see “Attribute Masking” on page 9).

If OGNL expressions might be used to map derived values into outgoing assertions and you want those values masked, select the related checkbox under the Attribute list (see “Using Attribute Mapping Expressions” on page 353).

9. Click Next.

10. On the Summary screen, review the configuration and click Done.

You can also click any heading to go back and change information.

11. On the Manage Adapter Instances screen, click Save.

Important: You must click Save if you wish to retain the adapter configuration.

Configuring the SP LDAP Adapter

1. If you have not already done so, establish a connection between PingFederate and your LDAP server (see “Configuring an LDAP Connection” on page 77).

2. Click SP Adapters on the Main Menu.

3. On the Manage Adapter Instances screen, click Create New Adapter Instance.

4. On the Type screen, enter an Instance Name and Instance Id, select LDAP Authentication Service 2.0 as the Adapter Type, and click Next.

The Instance Id may not contain spaces or underscores.
5. On the Instance Configuration screen, enter the values for adapter configuration as described on the screen and click **Next**.

   **Note:** If you do not know the values to enter at this time, select the “LDAP Datastore” and enter placeholders for other entries in valid formats similar to those shown in the screen example below. You can return to this screen to enter the correct values at any time (see “Configuring SP Adapters” on page 194). Click **Manage Data Stores** if you have not established a connection to your LDAP server.

6. (Optional) Click **Show Advanced Fields** and change default settings as needed.

   Due to import control restrictions, the standard JRE distribution supports strong but not unlimited encryption. To use the strongest AES encryption, when
Appendix B
LDAP Adapter Configuration

permissible, download and install the appropriate version of “Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files” from the Sun download Web site (http://java.sun.com/javase/downloads).

Ping Identity recommends that users apply strong password policies for encryption of data passed via the adapter. There are many resources for determining what constitutes a strong password. For more information, refer to any of the following sites:

- http://www.securityfocus.com/infocus/1192
- http://www.cert.org/tech_tips/unix_configuration_guidelines.html#A

7. Click **Next**.

8. On the Actions screen, click **Download** under Action Invocation Link.

9. On the next screen, click **Export** and save the properties file.

   The properties must be available to your application. Refer to the User Guide for your Integration Kit for details.

10. Click **Next**.

11. (Optional) On the Extended Contract screen, configure additional attributes as needed for this adapter instance (see “Extending an Adapter Contract” on page 197).

12. On the Summary screen, review the configuration and click **Done**.

   You can also click any heading to go back and change information.

13. If this is a second adapter instance, click **Next** and map at least one target URL to an adapter instance (see “Mapping URLs to Adapter Instances” on page 198).

   For a third and subsequent adapter instances, this step is optional.

14. On the Manage Adapter Instances screen (or the Map URLs to Adapter Instances screen), click **Save**.

   **Important:** You must click **Save** if you wish to retain the adapter configuration.
Application Endpoints

These endpoints provide a means, via standard HTTP, by which external applications can communicate with the PingFederate server.

**Note:** Begin each URL with the fully qualified server name and port number of your IdP or SP PingFederate server: for example: https://pingidentity.com:9031/idp/startSSO.ping.

The SSO and SLO endpoints for an IdP and an SP include optional parameters which you can use to specify error pages that users will see in the event of an SSO or SLO failure. By default, PingFederate provides templates for these and other errors or conditions (see “Customizing User-Facing Screens” on page 52).

For either SP or IdP servers, a maintenance endpoint is also provided for administrators to verify that the server is running (see “Maintenance Endpoint” on page 343).

IdP Endpoints

The following sections describe PingFederate IdP endpoints, including the query parameters that each accepts or requires. These endpoints accept either the HTTP GET or POST methods.

**Important:** When the parameter TargetResource (or TARGET) is used and includes its own query parameters, the parameter value must be URL-encoded. Any other parameters that contain restricted characters (many SAML URNs, for example) also must be URL-encoded.

For information about URL encoding, see, for example, “HTML URL-encoding Reference” (www.w3schools.com/tags/ref_urlencode.asp).
/idp/startSSO.ping

This is the path used to initiate an unsolicited IdP-initiated SSO transaction during which a SAML response containing an assertion is sent to an SP. Typically, a systems integrator or developer creates one or more links to this endpoint in the IdP application or portal to allow users to initiate SSO to various SPs.

For information about allowing applications to retrieve configuration data from the PingFederate server over SOAP, see “Web Service Interfaces” on page 345.

The following table shows the HTTP parameters for this endpoint.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PartnerSpId</td>
<td>The federation ID of the SP to whom the SAML response containing an assertion should be issued. One of these parameters is required unless the federation ID can be derived from TargetResource or TARGET (see below)</td>
</tr>
<tr>
<td>or PARTNER</td>
<td></td>
</tr>
<tr>
<td>TargetResource</td>
<td>For SAML 2.0, the value of either parameter is passed to the SP as the RelayState element of a SAML response message. This is the PingFederate implementation of the SAML 2.0 indicator for a desired resource at the SP during IdP-initiated SSO.</td>
</tr>
<tr>
<td>or TARGET (optional)</td>
<td>For SAML 1.x, the value is sent to the SP as a parameter named TARGET.</td>
</tr>
<tr>
<td>Note: If this parameter is not provided in the URL, then the target resource should be specified in the administrative console (see “Configuring a Default URL and Error Message” on page 107).</td>
<td></td>
</tr>
<tr>
<td>InErrorResource</td>
<td>Indicates where the user is redirected after an unsuccessful SSO. If this parameter is not included in the request, PingFederate redirects the user to the SSO error landing page hosted within PingFederate (see “Customizing User-Facing Screens” on page 52).</td>
</tr>
<tr>
<td>(optional)</td>
<td></td>
</tr>
<tr>
<td>Binding (optional)</td>
<td>Indicates the binding to be used; allowed values are URIs defined in the SAML specifications. For example, the SAML 2.0 applicable URIs are:</td>
</tr>
<tr>
<td></td>
<td>urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact</td>
</tr>
<tr>
<td></td>
<td>urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST</td>
</tr>
<tr>
<td>When the parameter is not used, the default ACS URL configured for the SP-partner connection is used, unless an ACS index is specified (see the next parameter, ACSIdx).</td>
<td></td>
</tr>
<tr>
<td>ACSIdx (optional - SAML 2.0)</td>
<td>Specifies the index number of partner’s ACS (see “Setting Assertion Consumer Service URLs (SAML)” on page 148). Takes precedence over the Binding parameter if both are specified. If neither the binding nor index is specified in the call, the default ACS is used.</td>
</tr>
</tbody>
</table>
IdP Endpoints

/idp/startSLO.ping

This is the path used to initiate an IdP-initiated SLO (under SAML 2.0). Typically, a systems integrator or developer creates one or more links to this endpoint in the protected resources of their IdP application or portal to allow users to end their sessions at various SPs. This endpoint uses the local PingFederate session to determine which SPs have been issued an SSO assertion and sends them a SAML logout request.

The following table shows the HTTP parameters for this endpoint.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IdpAdapterId (optional)</td>
<td>Allows an application to call out what IdP adapter to use for authentication (in a configuration with multiple IdP adapters).</td>
</tr>
<tr>
<td>RequestedFormat (optional - SAML 2.0)</td>
<td>Allows control over the NameId format.</td>
</tr>
</tbody>
</table>

/ idp/writecdc.ping

This endpoint is used for SAML 2.0 IdP Discovery. This is the path used when the IdP wants to write to the Common Domain Cookie (CDC) held within the user’s browser. The information written to the cookie indicates from which IdP this user has authenticated.
Appendix C
Application Endpoints

The following table shows the one HTTP query parameter for this endpoint.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TargetResource (optional)</td>
<td>Indicates where the user is redirected after successful IdP Discovery. If this parameter is not included in the request, PingFederate redirects the user to the referrer in the HTTP header. If there is no TargetResource or referrer, the call to this endpoint will fail.</td>
</tr>
</tbody>
</table>

/pf/heartbeat.ping

See “Maintenance Endpoint” on page 343.

SP Endpoints

The following sections describe the PingFederate SP endpoints, including the query parameters that each accepts or requires. These endpoints accept either the HTTP GET or POST methods.

Important: When the parameter TargetResource is used and includes its own query parameters, the parameter value must be URL-encoded. For information about URL encoding, see, for example, “HTML URL-encoding Reference” (http://www.w3schools.com/tags/ref_urlencode.asp).

/sp/startSSO.ping

This is the path used to initiate SP-initiated SSO. In this scenario, the SP issues an SSO request to the IdP asking for an SSO authentication response. Typically, a systems integrator or developer creates one or more links to this endpoint in SP applications to allow users to access various protected resources via SSO using the IdP as an authentication authority.

For information about allowing applications to retrieve configuration data from the PingFederate server over SOAP, see “Web Service Interfaces” on page 345.

The following table shows the HTTP parameters for this endpoint.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PartnerIdpId (required if more than one IdP connection is configured and Domain is not used)</td>
<td>The federation ID of the IdP that will authenticate the user and issue an assertion.</td>
</tr>
<tr>
<td>Domain (required to invoke Auto-Connect)</td>
<td>The domain name associated with the requesting user’s IdP (see “Using Auto-Connect” on page 15). In this case, PartnerIdpId cannot be used.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>TargetResource</strong> or <strong>TARGET</strong> (optional)</td>
<td>This parameter indicates where the end-user is redirected after a successful SSO. <strong>Note:</strong> If this parameter is not provided in the URL, then the target resource should be specified in the administrative console (see “Configuring Default URLs” on page 200).</td>
</tr>
<tr>
<td><strong>Binding</strong> (optional)</td>
<td>Indicates the binding to be used; allowed values are URIs defined in the SAML specifications. For example, the SAML 2.0 applicable URIs are:</td>
</tr>
<tr>
<td></td>
<td>urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact</td>
</tr>
<tr>
<td></td>
<td>urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST</td>
</tr>
<tr>
<td></td>
<td>urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect</td>
</tr>
<tr>
<td></td>
<td>When the parameter is not used for SAML 2.0, the first SSO Service URL configured for the IdP-partner connection is used (see “Specifying SSO Service URLs (SAML)” on page 235).</td>
</tr>
<tr>
<td><strong>InErrorResource</strong> (optional)</td>
<td>This parameter indicates where the end-user is redirected after an unsuccessful SSO. If this parameter is not included in the request, PingFederate redirects the user to the SLO error landing page hosted within PingFederate (see “Customizing User-Facing Screens” on page 52).</td>
</tr>
<tr>
<td><strong>SpSessionAuthnAdapterId</strong> (optional)</td>
<td>The explicit SP adapter instance ID indicating the adapter to use to create an authenticated session or security context.</td>
</tr>
<tr>
<td><strong>ForceAuthn</strong> (optional - SAML 2.0)</td>
<td>This parameter controls the attribute of the same name in the AuthnRequest. (The default is false.)</td>
</tr>
<tr>
<td><strong>IsPassive</strong> (optional - SAML 2.0)</td>
<td>This parameter controls the attribute of the same name in the AuthnRequest. (The default is false.)</td>
</tr>
<tr>
<td>** AllowCreate** (optional - SAML 2.0)</td>
<td>Controls the value of the AllowCreate attribute of the NameIDPolicy element in the AuthnRequest. (The default is true.)</td>
</tr>
<tr>
<td><strong>RequestedFormat</strong> (optional - SAML 2.0)</td>
<td>Specifies the value for the Format attribute in the NameIDPolicy element of the AuthnRequest. If not specified, the attribute is not included in the AuthnRequest.</td>
</tr>
<tr>
<td><strong>RequestedACSIdx</strong> (optional - SAML 2.0)</td>
<td>The index number of your site’s Assertion Consumer Service, where you want the assertion be sent to be sent.</td>
</tr>
<tr>
<td><strong>RequestedBinding</strong> (optional - SAML 2.0)</td>
<td>Indicates the binding requested for the response containing the assertion; allowed values are URIs defined in the SAML specifications.</td>
</tr>
</tbody>
</table>
Appendix C
Application Endpoints

If an adapter is specified in `SpSessionAuthnAdapterId`, then that adapter is used to create an authenticated session for SP-initiated SSO. If there is no `SpSessionAuthnAdapterId`, the ultimate destination of the user after SSO (either the `TargetResource` or the default SSO success URL) is used along with the mappings defined in the administrative console on the `Map URLs to Adapter Instances` screen (see “Mapping URLs to Adapter Instances” on page 198).

Note that adapter selection for SP-initiated SSO is similar to that for IdP-initiated SSO except that, because the adapter ID is dependent on the SAML deployment, PingFederate cannot expect it from an IdP. Therefore, it uses only the URL mapping for adapter selection for SSO.

### /sp/startSLO.ping

This is the path used to initiate SP-initiated SLO. Typically, a systems integrator or developer creates one or more links to this endpoint in the protected resources of their SP application, which allows users to end a session by sending a logout request to the IdP that authenticated the session.

Note that the IdP might send additional logout request messages to other SPs when it receives a logout request from a PingFederate server acting as an SP.

The following table shows the HTTP parameters for this endpoint.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>TargetResource</code> (optional)</td>
<td>Indicates where the user is redirected after a successful SLO. If this parameter is not included in the request, PingFederate uses as a default the URL for a successful SLO, as entered on the SP Default URLs screen.</td>
</tr>
</tbody>
</table>
An SP PingFederate session can be associated with one or more application sessions relying on any number of IDPs as the session authority. PingFederate must choose one session to terminate and also send an SLO request to the IDP that issued the assertion that created the session. Sessions are associated with the ID of the adapter instance that created them. Once an adapter ID is determined, the first session found with that ID is used. Determination of the adapter instance ID occurs in the following order:

1. If there is a value for the `SpSessionAuthnAdapterId` parameter, it is used.
2. If there is a value for the `SourceResource` parameter, PingFederate attempts to map a URL to an adapter using that value to determine the adapter ID.
3. If there is an HTTP header value for `Referer` [sic], PingFederate attempts to map a URL to an adapter using that value to determine the adapter ID.
4. If none of the above is successful, the `TargetResource` parameter value or the value for the default SLO success URL are used to map a URL to an adapter.
5. Finally, if no adapter ID is determined, the first one in the list is used.

### /sp/defederate.ping

This is the path used to terminate an account link created during SSO. Account linking provides a means for subject identification on the SP side. Links are created and terminated entirely by a user on the SP side. The link contains the name

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Binding (optional - SAML 2.0)</strong></td>
<td>Indicates the binding to be used; allowed values are URLs defined in the SAML specifications. The SAML 2.0 applicable URLs are:</td>
</tr>
<tr>
<td></td>
<td><code>urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact</code></td>
</tr>
<tr>
<td></td>
<td><code>urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST</code></td>
</tr>
<tr>
<td></td>
<td><code>urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect</code></td>
</tr>
<tr>
<td></td>
<td><code>urn:oasis:names:tc:SAML:2.0:bindings:HTTP-SOAP</code></td>
</tr>
<tr>
<td></td>
<td>When the parameter is not used, the first SLO Service URL configured for the IDP-partner connection is used (see “Specifying SLO Service URLs (SAML 2.0)” on page 237).</td>
</tr>
<tr>
<td><strong>InErrorResource (optional)</strong></td>
<td>Indicates where the user is redirected after an unsuccessful SLO. If this parameter is not included in the request, PingFederate redirects the user to the SLO error landing page hosted within PingFederate (see “Customizing User-Facing Screens” on page 52).</td>
</tr>
<tr>
<td><strong>SpSessionAuthnAdapterId (optional)</strong></td>
<td>The SP adapter instance ID indicating which session to terminate and which IDP will receive the logout request.</td>
</tr>
<tr>
<td><strong>SourceResource (optional)</strong></td>
<td>A URL indicating the origin of the logout request. It is mapped to an adapter ID in order to designate which session to terminate.</td>
</tr>
</tbody>
</table>
identifier from the IdP, the IdP’s federation ID, the adapter instance ID, and the local user identifier.

There are no HTTP parameters for this endpoint.

You can unlink a user session only if was established during SSO using an existing account link on the SP side. If more than one SP session was established via account linking on the same PingFederate session, each of those links will be terminated by this endpoint. A local logout is also performed for any link that is terminated.

/sp/cdcstartSSO.ping

This endpoint is used for IdP-Discovery implementations (see “IdP Discovery” in the “Supported Standards” chapter of Getting Started). This endpoint is similar to /sp/startSSO.ping and accepts the same parameters, with the exception of PartnerIdpId (see “/sp/startSSO.ping” on page 338). Instead of this parameter, the server attempts to use the common domain cookie to determine the IdP.

/sp/startAttributeQuery.ping

This endpoint is used to initiate an Attribute Query with a SAML 2.0 IdP (see “Attribute Query and XASP” in the “Supported Standards” chapter of Getting Started).

The following table shows the HTTP parameters for this endpoint.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subject</td>
<td>Uniquely identifies the user to the IdP. When user authenticates with an x.509 certificate, this is the Subject DN, which must be URL-encoded.</td>
</tr>
<tr>
<td>Issuer (optional)</td>
<td>The IssuerDN from the user’s x.509 certificate (when XASP is used), which uniquely identifies the entity that issued the user’s certificate. The parameter must be URL-encoded.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong>: When specified this parameter overrides the Subject parameter.</td>
</tr>
<tr>
<td>PartnerIdpId (except for XASP)</td>
<td>Used to identify the specific IdP partner to which the Attribute Query should be sent. If this parameter is not present, the Subject and Issuer are used to determine the correct IdP.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong>: For XASP, this parameter overrides both the Subject and Issuer parameters.</td>
</tr>
</tbody>
</table>
## Maintenance Endpoint

For either an SP or an IdP PingFederate server, an endpoint is provided to report whether the server is running.

### /pf/heartbeat.ping

This endpoint returns an “OK” browser message and an HTTP 200 status indication if the PingFederate server is running. If you receive an HTTP 404 error, the server associated with the endpoint is down.

Load balancers can use this endpoint to determine the status of PingFederate independently of checks used to determine the status of the supporting hardware.

You can also configure the server to provide regular status information to a network-management utility (see “Configuring Runtime Reporting” on page 60).

### Parameter Description

<table>
<thead>
<tr>
<th>Parameter (required for XASP, otherwise optional)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Format</td>
<td>Identifies the name-identifier format of the Subject query parameter. If included, the value must be one of the SAML 2.0 Name Identifier Format URIs (see section 8.3 of the SAML specifications (<a href="http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf">http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf</a>)). <strong>Note</strong>: For XASP, this parameter must be set to: <code>urn:oasis:names:tc:SAML:1.1:nameid-format:X509SubjectName</code> If not specified, the parameter defaults to: <code>urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified</code>. <strong>Note</strong>: The parameter must be URL-encoded.</td>
</tr>
<tr>
<td>AppId</td>
<td>The unique identifier of the initiating application.</td>
</tr>
<tr>
<td>SharedSecret</td>
<td>Used to authenticate the initiating application. The AppId and SharedSecret must both match the application authentication settings within the PingFederate server.</td>
</tr>
<tr>
<td>RequestedAttrName (optional)</td>
<td>A name of a user attribute requested from the IdP. For each such desired user attribute, include this parameter. If this parameter is not present, then all allowable user attributes are returned from the IdP.</td>
</tr>
</tbody>
</table>

See the next section, “Maintenance Endpoint”.

### Maintenance Endpoint

For either an SP or an IdP PingFederate server, an endpoint is provided to report whether the server is running.
Web Service Interfaces

PingFederate provides two Web Services for external clients. These services may be used by client applications to manage partner connections and support integration of Web applications, respectively:

- **Connection Management Service** — Enables creation and deletion of single connection configurations in PingFederate. This service may be used to migrate connections from one server environment to another (for example, from testing or staging to production) or to create new connections in a single server programmatically.

  **Tip:** PingFederate provides a command-line utility that can be used to export and modify connections, as well as other administrative-console configurations, and then import them to target environments (see “Automating Configuration Migration” on page 46).

- **SSO Directory Service** — Provides Web application developers with information regarding partner connections and adapter instances.

  **Tip:** Applications accessing the Connection Management Service must first authenticate themselves to the PingFederate server. SSO Directory Service authentication is optional by default, but may be required. For more information, see “Application Authentication” on page 95.

### Connection Management Service

The Connection Management Service supports basic connection management capabilities and is accessible only on a PingFederate server running the administrative console. This feature is useful in a variety of circumstances, but the following primary use cases were considered:
• As a utility to migrate changes to a partner connection though staging environments (for example: development, test, production).
  Changes to URLs and keys may be needed to make the connection appropriate to the next environment.
• As a way for an external application to update or delete connections programmatically, or create new ones using an exported connection XML file as a template.

The WAR file for this service, \texttt{pf-mgmt-ws.war}, is located in the \texttt{pingfederate/server/default/deploy2} directory.

\textbf{Note:} If you do not want to allow use of the service, it should not be deployed: remove the WAR file from the \texttt{deploy2} directory.

The SOAP-accessible service endpoint is \texttt{pf-mgmt-ws/ws/ConnectionMigrationMgr}.

The Web Services Description Language (WSDL) document describing this service can be retrieved from:
\url{https://<host_server>:<admin_console_port>/pf-mgmt-ws/ws/ConnectionMigrationMgr?wsdl}

\section*{Exporting a Connection}

You can export a connection either manually, using the administrative console, or programmatically, via a call to the Connection Management Service.

In either case, the exported XML complies with the standard SAML 2.0 metadata format, with extensions to capture PingFederate's proprietary configuration. Most connection configuration information is contained in the XML markup, with the exception of global configuration items such as adapter instances, data stores, and keypairs. Adapter instances and data stores are referenced by ID, and keypairs are referenced by the MD5 fingerprint of their X.509 certificate. Public certificates, such as the partner's signature verification certificate, are included completely (base-64 encoded).

\section*{Exporting Manually}

For information about using the administrative console to export SP connections at an IdP site, see “Using the Manage Connections Screen” on page 111.

For information about exporting IdP connection at an SP site, see “Using the Manage Connections Screen” on page 208.

\section*{Using the Connection Service}

The Connection Web Service exposes the following method for exporting connections:

\begin{verbatim}
public string getConnection(
    String entityId,
    String role,) throws IOException

Code Sample
Service service = new Service();
Call call = (Call)service.createCall();
call.setUsername("username");
call.setPassword("password");
\end{verbatim}
Importing Connections

Moving a connection from one PingFederate server to another requires care, as the target server must contain the global configuration items (data stores, keypairs, and adapter instances) that the connection references. Changing the references in the XML file—either manually or programmatically—may be necessary to adjust the connection to the target PingFederate environment.

Once required changes are made to the XML file, developers can use the Connection Management Service to import the connection into a different instance of PingFederate.

Tip: Alternatively, you can import XML connection files into PingFederate manually by copying them into the directory:

```xml
<pf_install>/pingfederate/server/default/data/connection-deployer
```

PingFederate scans this directory periodically and imports connections automatically.

Caution: Manually importing a connection always overwrites an existing connection with the same ID (the Web Service provides a switch to disallow this behavior, if desired—see below).

The Web Service exposes the following method for importing connections:

```java
public void saveConnection(
    String xml,
    boolean allowUpdate) throws IOException
```

The `xml` parameter is the complete representation of the connection retrieved by your application from an exported connection file (and optionally modified).

If `allowUpdate` is false, the Web Service can be used only to add a new connection. An error occurs if a connection already exists with the same connection ID and federation protocol in the XML. If `allowUpdate` is true and the connection already exists, it will be overwritten.

Code Sample

Below is example client code using the Apache AXIS libraries that invokes this Web Service to create a new connection:

```java
Service service = new Service();
Call call = (Call) service.createCall();
call.setTargetEndpointAddress("https://localhost:9999/pf-mgmt-ws/ws/ConnectionMigrationMgr");
call.setOperationName("getConnection");
Object result = call.invoke(new Object[]{"entityId", "SP"});
```

```java
public void saveConnection(
    String xml,
    boolean allowUpdate) throws IOException
```

```xml
<EntityDescriptor entityID="some_entity_id"
...>
</EntityDescriptor>
```

```
boolean allowUpdate = false;
```
Deleting Connections

The Web Service exposes the following method for connection deletion:

```java
public void deleteConnection(
    String entityId,
    String role) throws IOException
```

The `entityId` parameter is the Connection ID, which identifies the connection to be deleted. The `role` parameter is the connection role—IDP or SP.

Code Sample

Below is example client code using the Apache AXIS libraries that invokes this Web Service to delete a connection:

```java
Service service = new Service();
Call call = (Call) service.createCall();
call.setUsername("username");
call.setPassword("password");
call.setTargetEndpointAddress("https://localhost:9999/pf-mgmt-ws/ws/ConnectionMigrationMgr");
call.setOperationName("deleteConnection");
call.invoke(new Object[]{"entityid", "SP"});
```

Cluster Configuration Replication

A Web Service endpoint is available to replicate the administrative-console configuration to other nodes in a PingFederate cluster. This allows a client of this Web Service to create or update a new connection (or delete a connection) and then push the new configuration to the other cluster nodes.

The service endpoint is:

```
/pf-mgmt-ws/ws/ConfigReplication
```

The WSDL document describing this service can be retrieved from:

https://<host_server>:<admin_console_port>/pf-mgmt-ws/ws/ConfigReplication?wsdl

The Web Service exposes the following method:

```java
public void replicateConfiguration();
```

Code Sample

Below is example client code using the Apache AXIS libraries that invokes the configuration replication functionality:

```java
Call call2 = (Call) service.createCall();
call2.setUsername("joe");
call2.setPassword("test");
String addr2 = "https://localhost:9999/pf-mgmt-ws/ws/ConfigReplication";
call2.setTargetEndpointAddress(addr2);
call2.setOperationName("replicateConfiguration");
call2.invoke(new Object[]{});
```
Validation Disclaimer

The import process is not subject to the same rigorous data validation performed by the administrative user interface. Although some checks are made, it is possible to create invalid connections using the connection-migration process. Therefore, because the XML is complex and validation is limited, attempting to create an XML connection from scratch is not recommended. Rather, the administrative console should be used to create the initial connection. That way, changes necessary to the exported connection’s XML representation can be held to a minimum, reducing the risk of compromising data integrity.

SSO Directory Service

PingFederate SSO Directory Service allows applications to retrieve configuration data from a runtime PingFederate server. (A PingFederate server in a cluster configured as an administrative console does not support this Web Service.) This service allows Web applications to avoid storing and maintaining the data locally. These types of data can be retrieved:

- A list of IdP partners
- A list of SP partners
- A list of IdP adapter instances
- A list of SP adapter instances

The SSO Directory Service provides information useful for integrating an application with a PingFederate server. It is a way for the application to find out dynamically which partners can be used for SSO. This means applications need not be modified when new partners are configured in PingFederate.

The WAR file for this module, pf-ws.war, is located in the pingfederate/server/default/deploy directory.

Note: If you do not want to allow use of the service, it should not be deployed: remove the WAR file from the deploy directory.

The service endpoint is pf-ws/services/SSODirectoryService.

The WSDL document describing this service can be retrieved from:
http(s)://<pf_runtime_host>:<runtime_port>/pf-ws/services/SSODirectoryService?wsdl

You can retrieve a list using any of the following methods:

- getIdPList – Returns a list of active IdP connections configured for SP-initiated SSO. The list contains each IdP’s Connection ID and Connection Name
- getSPList – Returns a list of active SP connections configured for IdP-initiated SSO. The list contains each SP’s Connection ID and Connection Name

Note: For either IdP or SP lists, Connection IDs are returned as values for the XML tag <entityId>. Connection Names are returned as values for the XML tag <company> (see “SOAP Request and Response Example” on page 351).

- getAdapterInstanceList – Returns a list of SP adapter instances containing an ID and name.
• `getIdpAdapterInstanceList` – Returns a list of IdP adapter instances containing an ID and name.

**Note:** These methods do not require input parameters.

The service is also available over HTTP. The query string for retrieving any of the lists is:
/pf-ws/services/SSODirectoryService?method=<method_name>

### Code Example

When you integrate a Web application with PingFederate, use the SSO Directory Service to generate a connection or adapter list. The code needed to create any of the lists is similar.

The following Java code example retrieves an IdP list from the Web Service. The program calls the `getIdPList` method in the SSO Directory Service to retrieve an IdP list and print it to the console. This example uses the Apache Axis library and includes optional code for authentication to the PingFederate server (see “Application Authentication” on page 95). We recommend the use of HTTPS when including credentials.

```java
import org.apache.axis.client.Call;
import org.apache.axis.client.Service;
import java.net.URL;
import javax.xml.namespace.QName;
import com.pingidentity.ws.SSOEntity;
public class SSODirectoryClientSample {
    public static void main(String[] args) throws Exception {
        Service service = new Service();
        Call call = (Call) service.createCall();
        call.setUsername("username");
        call.setPassword("pass");
        URL serviceUrl = new URL(
                "https://localhost:9031/pf-ws/services/
                SSODirectoryService");
        QName qn = new QName("urn:BeanService", "SSOEntity");
        call.registerTypeMapping(SSOEntity.class, qn,
                new org.apache.axis.encoding.ser.BeanSerializerFactory(
                        SSOEntity.class, qn),
                new org.apache.axis.encoding.ser.BeanDeserializerFactory(
                        SSOEntity.class, qn));
        call.setTargetEndpointAddress( serviceUrl );
        call.setOperationName( new QName(
                "http://www.pingidentity.com/
                servicesSSODirectoryService",
                "getIDPList"));
        Object result = call.invoke( new Object[] {} );
        if (result instanceof SSOEntity[]) {
            SSOEntity[] idpArray = (SSOEntity[])result;
```
for (SSOEntity idp : idpArray)
{
    System.out.println(idp.getEntityId() + " " + idp.getCompany());
}
else
{
    System.out.println("Received problem response from server: " + result);
}

SOAP Request and Response Example

A client application must send a SOAP request to the PingFederate server specifying the requested Web Service and the specific method. For example, the following is a typical SOAP request for an IdP list using the SSO Directory Service.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope
    xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <soapenv:Body>
        <ns1:getIDPList
            soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
            xmlns:ns1="https://localhost:9031/ssodir/services/SSODirectoryService">
        </ns1:getIDPList>
    </soapenv:Body>
</soapenv:Envelope>
```

The PingFederate server’s Web Service will return a response containing the list you requested. The following is an example of a typical SOAP response for an IdP list:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <soapenv:Body>
        <getIDPListResponse
            soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
            <getIDPListReturn
                soapenc:arrayType="ns1:IDP[2]" xsi:type="soapenc:Array"
                xmlns:ns1="urn:BeanService"
                xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding">
                <getIDPListReturn href="#id0" />
                <getIDPListReturn href="#id1" />
            </getIDPListReturn>
        </getIDPListResponse>
    </soapenv:Body>
</soapenv:Envelope>
```
<getIDPListResponse>
  <multiRef id="id0" soapenc:root="0">
    <soapenv:encodingStyle=
      "http://schemas.xmlsoap.org/soap/encoding/"
    xsi:type="ns2:IDP"
    xmlns:soapenc=
      "http://schemas.xmlsoap.org/soap/encoding/"
    xmlns:ns2="urn:BeanService">
      <company xsi:type="xsd:string">MegaMarket</company>
      <entityId xsi:type="xsd:string">www.megamarket.com</entityId>
    </multiRef>
    <multiRef id="id1" soapenc:root="0">
      <soapenv:encodingStyle=
        "http://schemas.xmlsoap.org/soap/encoding/"
      xsi:type="ns3:IDP" xmlns:ns3="urn:BeanService"
      xmlns:soapenc=
        "http://schemas.xmlsoap.org/soap/encoding/">
        <company xsi:type="xsd:string">Ping</company>
        <entityId xsi:type="xsd:string">pingfederate3:default:entityId</entityId>
      </multiRef>
  </soapenv:Body>
</soapenv:Envelope>
Using Attribute Mapping Expressions

PingFederate provides an advanced option allowing administrators to map user attributes by way of an expression language. Because the option carries with it a potential for misuse, however, it is disabled in the administrative console for security reasons.

Tip: If you are upgrading to PingFederate 5.1 or higher and importing a configuration archive that uses expression mapping, the feature will be enabled automatically.

This appendix describes the option, which is based on the Object-Graph Navigation Language (OGNL), and how to enable or disable it.

Caution: The security concern posed by OGNL is related to a potential for abuse by PingFederate administrative users within an organization; the concern is not related to any known external threats. We recommend, however, that the option be enabled only if required.

About OGNL

OGNL is based on the Java programming language. OGNL expressions are useful for evaluating and manipulating attribute values and returning information based on the results. You can also transform a range of values into a text description, or do the same for a sequence of ranges.

In the expression below, for example, the value of the attribute “net-worth” is transformed first to eliminate any dollar signs or commas, then the result is evaluated to determine whether the user’s net worth falls into a “bronze,” “silver,” or “gold” category:

```
#result=#this.get("salary").toString(),
#result=#result.replace("$","")
#result=#result.replace(",","")
#result < 500000 ? "bronze" :
```
#result < 1000000 ? "silver" : "gold"

Use the # symbol to reference OGNL variables. For an IdP, PingFederate provides predefined OGNL variables for IdP-adapter attributes as well as any attributes retrieved from data stores. For an SP, variables are available for attributes received in an assertion or an attribute query. For example, the SAML_SUBJECT value may be retrieved using:

```java
#SAML_SUBJECT
```

Important: Because OGNL uses the “at” symbol (@) to reference static Java methods, expressions containing the symbol must be enclosed in double quotes; otherwise, expression parsing will fail. For example:

```java
#SAML_SUBJECT="usr@msn.com"
```

For data-store attributes, use this syntax (for example):

```java
#this.get("ds.amount")
```

For more information, see “Using the OGNL Edit Screen” on page 355.

For more information about OGNL, see the OGNL Web site, www.ognl.org.

Note: The PingFederate runtime engine uses OGNL version 2.6.7

### Enabling/Disabling Expressions

OGNL can be manually enabled or disabled for attribute mapping by editing a configuration file located in:

```
<pf_install>/pingfederate/server/default/data/config-store/
```

Important: If OGNL is enabled and expressions configured anywhere in the administrative console, disabling the feature will cause errors during runtime processing.

To enable or disable OGNL expressions:

1. In the directory cited above, open the file:

   ```
   org.sourceforge.common.ExpressionManager.xml
   ```

2. Change the value of the element named evaluateExpressions to either true or false and save the file. For example:

   ```
   <item name="evaluateExpressions">true</item>
   ```

Note: The absence of a value (the installed default) does not necessarily disable the use of OGNL expressions. To facilitate backward compatibility, when no value is present, configuration archives containing expressions can be imported successfully, and further use of the feature is enabled. (The term “silent” is used for this condition in the server log.)
3. Start or restart PingFederate.

**Tip:** If you are enabling OGNL to use for mapping SaaS-provisioning attributes, it is not necessary to restart the PingFederate server.

When OGNL expressions are enabled, the selection Expression is available in the drop-down menus under Source in each of the administrative-console Fulfillment screens (see Figure 5), and the feature is available on the SaaS-provisioning attribute-mapping screen (when SaaS Provisioning itself is enabled—see “SaaS Provisioning” on page 18).

When you make this selection, you can enter the expression in the text field provided. You can also test expressions (see the next section).

### Using the OGNL Edit Screen

An in-line editor is available for OGNL expressions. The editor validates the expression and allows an administrator to enter input values and test the resulting output.

- To reach the OGNL editor, click **Edit** under Actions for an expression on any of the attribute Fulfillment screens.
Appendix E

Using Attribute Mapping Expressions

Here is an example of the edit screen, from the IdP configuration flow:

To test an expression:

1. Enter an input value in the Value text box associated with the attribute.
2. Click the Test link near the bottom right of the screen.

If the expression contains no errors, the result is displayed under Test Results.

Important: If you make changes to an expression and want to save them, click Update under Actions. To discard changes, click the Cancel link under Actions; clicking the Cancel button near the bottom of the screen discards all changes you have made in the current task.
Basic troubleshooting tips are provided here to help overcome common difficulties. Please contact Ping Identity at 877.898.2905 or support@pingidentity.com for further information.

This appendix contains the following sections:

- “Data Stores” on page 357
- “Installation” on page 358
- “Protocols” on page 358
- “Server” on page 358

## Data Stores

### Table 13: Troubleshooting Data Stores

<table>
<thead>
<tr>
<th>Problem</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>When setting up the JDBC data store, a connection cannot be established.</td>
<td>Verify that the proper drivers and connectors have been installed. Also, verify the connection URL, username, and password. If unsuccessful, contact your database administrator.</td>
</tr>
</tbody>
</table>
Table 13: Troubleshooting Data Stores (Continued)

<table>
<thead>
<tr>
<th>Problem</th>
<th>Solution</th>
</tr>
</thead>
</table>
| Cannot connect to a Directory Service with the LDAP protocol.          | Verify the connection URL, port, principal, and credentials. If unsuccessful, contact your system administrator. If using LDAP with SSL/TLS (ldaps://), ensure the LDAP server’s SSL certificate is signed by a trusted certificate authority or import the certificate into your JAVA_HOME/jre/lib/security/cacerts keystore (consult your Java documentation and the Java keytool documentation).
|                                                                        | The trusted certificates stored and accessed through the PingFederate console are used only with SOAP connections, not other SSL/TLS connections.                                                     |

Installation

Table 14: Troubleshooting Installation

<table>
<thead>
<tr>
<th>Problem</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error message “Not enough memory on the server”</td>
<td>Verify that there is at least 1,024 MB of RAM installed on the server (see “System Requirements” in the “Installation” chapter of Getting Started).</td>
</tr>
</tbody>
</table>

Protocols

Table 15: Troubleshooting Protocols

<table>
<thead>
<tr>
<th>Problem</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Certificates unexpectedly expire.</td>
<td>Verify that the server clocks are synchronized on both sides of the federation. Note that you can configure PingFederate to notify administrators in advance of impending certificate expiration (see “Configuring Runtime Notifications” on page 59).</td>
</tr>
</tbody>
</table>

Server

Table 16: Troubleshooting the PingFederate Server

<table>
<thead>
<tr>
<th>Problem</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>PingFederate does not start.</td>
<td>Make sure that the Java SDK is installed (see “Installing the JDK” in the “Installation” chapter of Getting Started).</td>
</tr>
</tbody>
</table>
account link
A persistent name identifier that enables federation of separately established accounts among disparate domains (see also account linking and pseudonym).

account linking
A form of identity mapping among separate user accounts managed under different Internet domains. The mapping typically involves a name identifier—which may be a pseudonym—used to link the user to each account. The identifier is persisted at the SP site to enable seamless SSO/SLO. Additional attributes may be sent with the identifier.

account mapping
A form of identity mapping by which one or more user attributes is passed in a single sign-on transaction. The attributes are used at the destination site as a means identifying the user and looking up local account information.

adapter
Supplementary software that allows PingFederate to interact with Web applications and systems. Two adapter choices are bundled with PingFederate: an OpenToken Adapter for use with separately available developer integration kits, and an LDAP adapter for use with your active directory data store.

adapter contract
A list of attributes “hard-wired” to an adapter and conveyed generally via cookies between the adapter and application.

artifact
A reference to a SAML protocol message. The federation partner that receives the artifact dereferences it, identifying the sender, and requests the complete message in a separate SOAP transaction.

Artifact Resolution Service
The SOAP endpoint that processes artifacts returned from a federation partner to retrieve the referenced XML message. Can be used to dereference authentication requests, assertion responses, and SLO messages.

assertion
A SAML XML document that contains identifying information about a particular subject; i.e., a person, company, application, or system. A SAML assertion can contain authentication, authorization, and attribute information about the subject.

Assertion Consumer Service
A SAML-compliant portion of PingFederate in an SP role that receives and processes assertions from an IdP.

attributes
Distinct characteristics that describe a subject. If the subject is a Web site user, attributes may include a name, group affiliation, email address, etc.

attribute contract
A list of attributes, agreed to by the partners in an identity federation, representing information about a user (SAML subject). The attributes are sent from the IdP to the SP during SSO or STS processing.
**attribute mapping**
A form of identity mapping between IdP and SP user accounts that uses attributes to identify the user or provide supplemental information.

**audience**
The XML element in a SAML assertion that uniquely identifies a Service Provider.

**authentication context**
An element in a SAML assertion indicating the method or process used by an IdP to authenticate the subject of the assertion; may be used for authorization decisions or auditing compliance.

**attribute source**
Specific database or directory location containing data needed by an IdP to fulfill a connection partner’s attribute contract or by an SP to look up additional attributes to fulfill an adapter contract.

**back-channel**
Server-to-server, cross-domain communication path using a protocol, typically SOAP, that does not rely on a browser as an intermediary.

**binding**
A mapping of SAML request and response messages to specific transport protocols (redirect, POST, or artifact).

**certificate**
A digital file used for identity verification and other security purposes. The certificate, which is often issued by a Certificate Authority (CA), contains a public key, which can be used to verify the originator’s identity.

**Certificate Revocation List**
(CRL) A list of revoked signing certificates, maintained by the issuing authority at a public URL.

**channel**
A dedicated SaaS Provisioning configuration specific to a particular service partner, data source, and target service.

**connection partner**
Entities, such as companies, that are part of an identity federation. These entities are referred to as connection partners in the PingFederate configuration process.

**credential**
Information used to identify a subject for access purposes (e.g., username and password). A credential can also be a certificate.

**Database Management System**
A system for storing and maintaining user account information and attributes. The tables and columns in the RDBMS are used by PingFederate to create user look-up and attribute retrieval queries. (See Java Database Connectivity.)

**data store**
A database or directory location containing user account records and associated user attributes.

**defederation**
Optional user-initiated delinking of an identity federation that uses a persistent name identifier or pseudonym for account linking.

**digital signature**
A process for verifying the identity of the originator of an electronic document and whether the document has been intercepted or altered. The process involves message signing, signature validation, and signing policy coordination between partners.

**endpoint**
A terminal or gateway that generates or terminates a stream of information. For example, a PingFederate SP server contains an endpoint for the Assertion Consumer URL.

**entity ID**
The XML element in a SAML assertion that uniquely identifies an Identity Provider.

**Extensible Markup Language**
A structured, hierarchical text format—based on SGML (Standard Generalized Markup Language)—for the flexible and organized exchange of data.

**HTTP cookie**
Information sent from a server to a Web browser to identify a registered Web site user. Once the cookie is placed in the browser, it is sent back to the server to identify the user every time the user accesses the site. PingFederate’s integration adapters interface with the cookie.

**HTTP header**
The section of an HTTP request or response containing information about the client or the server. PingFederate can use HTTP headers to look up session information passed by the IdP’s Web application.

**HTTP request parameter**
A named parameter sent as part of a URL request from a browser to a Web server.
identity federation
A trust agreement between or among organizations, implemented using accepted standards, to provide user-authentication tokens and other user or system attributes securely across Internet domains, primarily to enable cross-domain SSO.

Identity Provider
The identity source or SAML authority that authenticates a subject and provides an SP with a security assertion vouching for that authentication.

IdP-initiated SSO or SLO
An identity federation transaction in which the initial action requiring a security context from an IdP occurs at a IdP’s site. For example, the user is logged on to the IdP and requests protected resources on an SP. The IdP sends authentication information to the SP.

inbound
A direction of message flow coming into a server relative to the server's identity federation role (IdP or SP). For an IdP, inbound messages include SAML authentication requests. For an SP, inbound messages include SAML assertions.

Java Database Connectivity (JDBC)
A Java API that allows Java programs to interact with databases.

dkeys
The length (in bits) of each key in a keypair.

keypair
The private key and public key represented by a certificate. PingFederate uses the private key of its keypair(s) to generate signatures for assertions, requests, and responses, as applicable.

Lightweight Directory Access Protocol
A set of protocols used for accessing information directories. PingFederate uses the LDAP v3 protocol for user look-up and attribute processing.

metadata
The SAML 2.0 standards define a metadata exchange schema for conveying XML-formatted information between two SAML entities. Metadata includes endpoint URLs, binding types, attributes, and security-policy information.

Online Certificate Status Protocol (OCSP)
A standard developed by the Internet Engineering Task Force that enables Internet applications to obtain the current status of signing certificates, indicating whether a certificate has been revoked, via HTTP.

opaque
Not readable. If a user's subject identifier is opaque, the an SSO partner cannot directly identify the user with reference to the source. An persistent identifier, or pseudonym, can be used for Account Linking.

outbound
A direction of message flow leaving a server. For an IdP, outbound messages include SAML assertions. For an SP, outbound messages include SAML authentication requests.

partner
See connection partner.

portal
A Web-based application, accessed using a Web browser, that often aggregates content from multiple providers and/or serves as a central point of entry.

POST
An HTTP method of transmitting data contained in HTML forms, by which the data appears in the message body.

principal
A user, system, or process whose identity can be authenticated. See subject.

profiles
Rules that describe how to embed SAML assertions into and extract them out of other protocols in order to enable SSO or SLO. Profiles describe SAML request and response flows that fulfill specific use cases.

protected resource
Information, typically accessed via a Web URL, that is protected by an access management system. See target URL.

protocol
An agreed-upon format for transmitting data. XML format of SAML request or response messages.

pseudonym
A persistent name identifier assigned to a user and shared among entities, usually with the user’s permission, to enable SSO and SLO. Pseudonyms are often used with the SAML account linking protocol to enable SSO while preventing the discovery of the user’s identity or activities.

Public Key Infrastructure (PKI)
Enables users of an unsecured public network, such as the Internet, to securely and privately exchange data and money through the use of keypairs and certificates. The PKI provides for a digital certificate that can identify an individual or an organization and
directory services that can store and, when necessary, revoke the certificates.

**redirect**
A SAML binding that conveys a request or response by sending the user's browser to another location. For instance, an authentication request can be sent from an SP through a browser to an IdP.

<RequestSecurityToken>
(RST) WS-Trust or WS-Federation XML element identifying a request for validation of a security token, or for validation and then issuance of a replacement security token.

<RequestSecurityTokenResponse>
WS-Trust or WS-Federation XML element identifying a response to an RST and containing either the status of the submitted security token or both the status and (if requested and the received token is valid) a newly issued token for further SSO or Web-Services processing.

**SAML**
See Security Assertion Markup Language.

**SAML authority**
A security domain that issues SAML assertions.

**Secure Sockets Layer**
An encryption protocol that sends data between a client and server over a secure HTTP connection.

**Security Assertion Markup Language**
(SAML) A standard, XML-based, message-exchange framework enabling the secure transmission of authentication tokens and other user attributes across Internet domains.

**security domain**
An application or group of applications that trust a common security token used for authentication, authorization, or session management. The token is issued to a user after the user has authenticated to the security domain.

**security token**
A collection of information used to establish acceptable identity for security purposes. Tokens can be in binary or XML format. A SAML assertion is one kind of security token.

**Security Token Service**
An entity responsible for responding to WS-Trust requests for validation and issuance of security tokens used for SSO authentication to Web Services.

**service-oriented architecture**
A loosely coupled application architecture in which all functions or services are accessible via standard protocols. Interfaces are platform and programming-language independent.

**Service Provider**
A system entity that provides access to a protected resource based on authentication information supplied by an IdP.

**SP-initiated SSO or SLO**
An identity-federation transaction in which the initial action requiring a security context from an IdP occurs at a SP’s site.

**session persistence**
A mechanism for identifying a user or browser for subsequent requests to a server, needed because the HTTP protocol is stateless. This information is used to lookup state information for the user—for example, items in a shopping cart. PingFederate does not implement session persistence; it facilitates the communication of session information between systems that do.

**Simple Object Access Protocol**
(SOAP) Defines the use of XML and HTTP to access services, objects, and servers in a platform-independent manner.

**Single Logout**
The process of logging a user out of multiple “session participants” or sites where the user has started an SSO session.

**Single Logout Return Service**
The SAML implementation endpoint URL that returns logout requests.

**Single Logout Service**
The SAML implementation endpoint URL that receives logout requests for processing.

**Single Sign-On**
(SSO) The process of authenticating an identity (signing on) at one Web site (usually with a user ID and password) and then accessing resources secured by other domains without re-authenticating.

**Single Sign-on Service**
The SAML implementation endpoint URL that receives authentication requests for processing.

**Source ID**
A 20-byte sequence used to determine an IdP’s identity.
subject
A person, computer system, or application. In the SAML context, assertions make statements about subjects. See principal.

target URL
The SP's protected resource; the end destination of an SSO event. See protected resource.

transient name identifier
A temporary ID used to preserve user anonymity while facilitating account linking.

token exchange
The process by which a security token is exchanged for another security token.

token translators
An aggregate term for both token processors (used by the IdP PingFederate Security Token Service (STS) to handle different types of incoming security tokens) and token generators (used by the SP PingFederate STS to issue various types of tokens).

Uniform Resource Identifier
Identifies an Internet resource with a string of characters conforming to a specified format.

Uniform Resource Locator
Identifies an Internet resource according to its Internet location.

virtual server ID
An optional unique identifier by which an identity federation deployment can be known to a specific connection partner.

Web Services Security
A standard mechanism for securing Web Service interactions, often by binding a security token to the Web Service request.

Web Services
Nonbrowser-based, loosely coupled applications that provide modular, programming-language-independent access to specific functions and data across the Internet, via XML and standard protocols.

Web Service Client
An entity that requests a Web Service interaction. In the context of an STS, the Web Service Client would request that a security token be issued for the interaction.

Web Service Enhancement
Supplemental software for the .NET framework provided by Microsoft.

Web Service Provider
In the context of an STS, an entity that requests validation of the security token sent with a client's request for service.

WS-SX
The OASIS committee working on WS-Trust.

WS-Trust
A standard protocol by which an application can request that an STS issue, validate, or exchange security tokens.
# List of Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACS</td>
<td>Assertion Consumer Service</td>
</tr>
<tr>
<td>API</td>
<td>Application Programmer Interface</td>
</tr>
<tr>
<td>ARS</td>
<td>Artifact Resolution Service</td>
</tr>
<tr>
<td>CA</td>
<td>Certificate Authority</td>
</tr>
<tr>
<td>CRL</td>
<td>Certificate Revocation List</td>
</tr>
<tr>
<td>CSR</td>
<td>Certificate Signing Request</td>
</tr>
<tr>
<td>DBMS</td>
<td>Database Management System</td>
</tr>
<tr>
<td>DMZ</td>
<td>Demilitarized Zone</td>
</tr>
<tr>
<td>DN</td>
<td>Distinguished Name (certificate identifier)</td>
</tr>
<tr>
<td>DNS</td>
<td>Domain Name System</td>
</tr>
<tr>
<td>EIM</td>
<td>Enterprise Identity Management</td>
</tr>
<tr>
<td>GUI</td>
<td>Graphical User Interface</td>
</tr>
<tr>
<td>HTTP</td>
<td>HyperText Transfer Protocol</td>
</tr>
<tr>
<td>HTTPS</td>
<td>Secure HyperText Transfer Protocol</td>
</tr>
<tr>
<td>IdM</td>
<td>Identity Management</td>
</tr>
<tr>
<td>IdP</td>
<td>Identity Provider</td>
</tr>
<tr>
<td>IP</td>
<td>Internet Protocol</td>
</tr>
<tr>
<td>J2SDK</td>
<td>Java 2 Software Development Kit</td>
</tr>
<tr>
<td>JDBC</td>
<td>Java Database Connectivity (JDBC)</td>
</tr>
<tr>
<td>LDAP</td>
<td>Lightweight Directory Access Protocol</td>
</tr>
<tr>
<td>O</td>
<td>Organization</td>
</tr>
<tr>
<td>OASIS</td>
<td>Organization for the Advancement of Structured Information Standards</td>
</tr>
<tr>
<td>OCSP</td>
<td>Online Certificate Status Protocol</td>
</tr>
<tr>
<td>OU</td>
<td>Organizational Unit</td>
</tr>
<tr>
<td>PKI</td>
<td>Public Key Infrastructure</td>
</tr>
<tr>
<td>RDBMS</td>
<td>Relational Database Management System</td>
</tr>
<tr>
<td>RST</td>
<td>&lt;RequestSecurityToken&gt;</td>
</tr>
<tr>
<td>RSTR</td>
<td>&lt;RequestSecurityTokenResponse&gt;</td>
</tr>
<tr>
<td>SAML</td>
<td>Security Assertion Markup Language</td>
</tr>
<tr>
<td>SaaS</td>
<td>Software as a Service</td>
</tr>
<tr>
<td>SDK</td>
<td>Software Development Kit</td>
</tr>
<tr>
<td>SP</td>
<td>Service Provider</td>
</tr>
<tr>
<td>SLO</td>
<td>Single Logout</td>
</tr>
<tr>
<td>SOA</td>
<td>service-oriented architecture</td>
</tr>
<tr>
<td>SOAP</td>
<td>Simple Object Access Protocol</td>
</tr>
<tr>
<td>SQL</td>
<td>Structured Query Language</td>
</tr>
<tr>
<td>SSL</td>
<td>Secure Sockets Layer</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>SSL/TLS</td>
<td>Secure Sockets Layer/Transport Level Security</td>
</tr>
<tr>
<td>SSO</td>
<td>Single Sign-On</td>
</tr>
<tr>
<td>SSTC</td>
<td>Security Services Technical Committee (of OASIS)</td>
</tr>
<tr>
<td>STS</td>
<td>Security Token Service</td>
</tr>
<tr>
<td>TCP</td>
<td>Transmission Control Protocol</td>
</tr>
<tr>
<td>URI</td>
<td>Uniform Resource Identifier</td>
</tr>
<tr>
<td>URL</td>
<td>Uniform Resource Locator</td>
</tr>
<tr>
<td>WSC</td>
<td>Web Service Client</td>
</tr>
<tr>
<td>WSP</td>
<td>Web Service Provider</td>
</tr>
<tr>
<td>WSS</td>
<td>Web Services Security</td>
</tr>
<tr>
<td>XASP</td>
<td>X.509 Attribute Sharing Profile</td>
</tr>
<tr>
<td>XML</td>
<td>Extensible Markup Language</td>
</tr>
</tbody>
</table>
Index

A
access control 34
account linking 5
    configuring 219
account mapping 5
accounts, administrator 34
activation
    for IdP connection 265
    for SP connection 182
adapters
    commercial 4
    configuring
        IdP 102
        SP 194
integration 4
mapping URLs to SP 198
administrative console
    certificate authentication to 38
    lockout recovery 35
    logging on using LDAP 38
administrators, adding 36
affiliations, SP 183
application endpoints
    IdP 108
    SP 202
archiving data 32
artifact
    lifetime, setting for IdP connections 239
    lifetime, setting for SP connections 152
Assertion Consumer Service URLs 148
assertions
    content 125
    lifetime 124
    mapping attributes to 131
attribute authority (XASP), configuring 252
attribute contract 7
    default for SP connection 145
    IdP connection 220
    to SP 127
attribute query
    configuring for IdP connection 252
    configuring for SP connection 157
    mapping names 253
attribute sources
    custom 79
    failover (for IdP) 133
    for IdP 133
attributes
    mapping
        about 5
        for SP connection 142, 159, 297, 316
        for user provisioning 250
    masking in log files 9, 25
    overview 6
    sources for express provisioning 245
audit logging 20
Auto-Connect
    about 15
    activation (IdP connection) 267
    activation (SP connection) 191
    allowed IdP domains 268
    allowed SP domains 192
    configuring (as an IdP) 189
    configuring (as an SP) 265
    metadata lifetime 71
    metadata signing 70
back-channel settings
  for IdP connections 255
  for SP connections 161
backups, administrative console 32
bindings
  allowable
    for IdP connection 238
    for SP connection 152
configuring
  for IdP connection 216
  for SP connection 121
SOAP for IdP connection 255
SOAP for SP connection 161
certificate authority (CA) 87
certificates
  authority, verifying 14
  considerations 19
digital signing 93
direction notification 59
for metadata 30
for SOAP authentication 19
management 10
revocation of 97
selecting XML decryption (SP-to-IdP) 264
selecting XML encryption (SP-to-IdP) 262
SSL 58, 88
SSL client 91
trust models for 12
verifying 14
checklist, for federation 19
common domains 82
configuration
  archive 32
  exporting 32
  importing 33
connection list
  for IdP connection 208
  for SP connection 111
connections
  importing 345
  migrating 46
templates for SP 186
cookies, common domain 82
cookies, for IdP Discovery 82
custom data stores
  filters (IdP connections) 232
  filters (SP connections) 142, 316
  selecting fields (IdP connections) 232
  selecting fields (SP connections) 142
data archiving 32
data exchange
  automated 21
    for IdP connection 21
    for SP connection 21
data store 20
  description 9
    for attribute query profile 158
    introduction 72
    JDBC configuration 73
    LDAP configuration 77
  selecting
    for express provisioning 69, 246
    for IdP connection 225
    for SaaS provisioning 174
    for SP connection 134
troubleshooting 357
database
  filter
    for IdP connection 227
    for IdP connection 226
    for SP connection 135
database filter
  for SP connection 137, 311
decryption keys
  for IdP connections 264
  for SP connections 169
defederation 5
digital signatures
  about 10
  keys and certificates 93
  policy 12
digital signing
  for IdP connection 258
  for metadata 30
  for SP connection 164
directory service
  code example 350
  SOAP example 351
  SSO 349
discovery, IdP 81
domains
  allowed for IdP Auto-Connect 268
  allowed for SP Auto-Connect 192
data store 20
email
  addresses 59
  notifications 59
server configuration 40
encryption, XML 155, 242
endpoints
  IdP application 108
protocol, IdP 108
protocol, SP 205
SP application 202
canada
entity ID 67
error message, SLO 107
event trigger, for provisioning 245

F
failover
for data stores (IdP) 133
failsafe attribute source 144
federation
   checklist 19
defederation 5
   ID 21
   roles, choosing 64
   URLs 67
filter, database
   for IdP connection 227
   for SP connection 137, 311
filter, LDAP
   for IdP connection 231
   for SP connection 140, 314

H
heartbeat, server
   checking via HTTP 343
host names, virtual 42

I
identity mapping
   about 5
   IdP configuration 125
   SP configuration 219
IdP
   adapters 102
   connections
      activation 265
      introduction 193
default URL 107
   SLO error message 107
IdP Discovery
   common domain service 82
   configuring 81, 82
   cookies for 82
   local domain service 84
   selecting 64
IdP-initiated SLO
   configuring
      for IdP connection 218
      for SP connection 123
IdP-initiated SSO
configuring
   for IdP connection 217
   for SP connection 122
importing
   metadata
      for IdP connection 211
importing connections 345
importing metadata
   for SP connection 116
installation
   license key 45
troubleshooting 358
integration kits 4

J
JDBC configuration 73

K
keys
   for XML decryption (SP-to-IdP) 264
   for XML encryption (IdP-to-SP) 169
keys and certificates
   digital signature 93
   SSL 91

L
LDAP
   configuration 77
directory search
      for SP connection 139, 313
   filter 140, 231, 314
   pooling options 79
   SSL 78
   using SSL 78
LDAP, using for authentication 38
license key, installing 45
licensing
   email address to notify 59
   expiration notification 59
Lifetime 71
Linux
   starting the server 24
   stopping the server 24
lockout, administrative console 35
log files 24
logging
   administrator audit 25
   files 24
   modes 27
   transaction 26
M
main menu
   for IdP connection 207
   for SP connection 110
mapping
   adapters
      URLs to SP 198
   attributes
      configuring for SP connection 142, 159, 297, 316
      identity 5, 125, 219
masking attributes 9
masking attributes in logs 25
message
   signing 10
   validation 11, 19
metadata
   exporting 27
   exporting XML encryption certificates 30
   importing
      for IdP connection 211
      for SP connection 116
   lifetime for Auto-Connect 71
   providing for Auto-Connect 15
   signing 31
   signing for Auto-Connect 70
   use 21
migrating connections 46
monitoring
   JMX 61
   provisioning 61
   server 60

N
name identifiers
   SAML 2.0 125
   WS-Federation 127
notifications
   runtime 59

O
Object-Graph Navigation Language (OGNL), editing 355

P
password notification 34
passwords
   changing 38
   resetting 37
planning checklist 19
pooling options
   database 75
   LDAP 79
profiles
   configuring
      for IdP connection 216
      for SP connection 121
   saving configuration
      for IdP connection 156, 243
protocol endpoints
   IdP 108
   SP 205
protocols
   choosing 64
   troubleshooting 358
provisioning
   enabling 216
   event trigger for 245
   mapping attributes for 250
provisioning, as SP
   choosing data stores for 246
provisioning, SaaS
   choosing internal data store 69
   choosing source data stores for 174
pseudonyms
   unique values for 106

R
requester mapping, for XASP 203
resetting passwords 37
roles, choosing IdP/SP 64
runtime
   configuration 42
   reporting 60

S
SAML
   selecting 64
SAML 1.1
   SP connection steps 121, 216
SAML 2.0
   SP connection steps 121, 215
SDK 4
security
   back channel 19
   considerations 10
server
   troubleshooting 358
server HTTP checking 343
server ID 20
server monitoring 60
server status verification 343
service URL
   IdP, for WS-Fed 236
   SP, for WS-Fed 150
session integration considerations 20  
Signing 70  
signing XML files 31  
SLO service URLs 150  
SNMP configuration 60  
SOAP responder URL 21  
SP  
   adapters 194  
   connections  
      activation 182  
      introduction 101  
      default URL 200  
SP-initiated SLO, configuring  
   for IdP connection 218  
   for SP connection 123  
SP-initiated SSO, configuring  
   for IdP connection 217  
   for SP connection 123  
SSL 10  
   about 13  
   certificates 88  
   keys and certificates 91  
   using for LDAP 78  
SSO directory service 349  
   standards  
      choosing 64  
starting the server  
   Linux 24  
   Windows 24  
stopping the server  
   Linux 24  
   Windows 24  
summary  
   for IdP connection 265  
   for SP connection 182  
synchronization  
   clock 20  
   provisioning 69  
system administration (single or multiple users) 58  

T  
time synchronization 20  
transaction logging 20, 26  
transport security considerations 19  
trigger, for provisioning 245  
troubleshooting 357  
   data store 357  
   installation 358  
   protocols 358  
   server 358  
   trust models 12  

U  
unique IDs 20  
unique values, for pseudonym creation 106  
upgrading PingFederate 32  
user management  
   setting 58  
   users  
      account management 34  
      adding 36  

V  
validating messages 19  
verifying certificates 14  
virtual host names 42  
virtual IDs 20  
   for IdP servers 111  
   for SP servers 208  

W  
Web Services  
   connection management 345  
   SSO Directory 349  
Web templates 52  
   where clause  
      for IdP connection 227  
      for SP connection 137, 311  
WS-Federation  
   selecting 64  
   setting realm ID 67  
   SP connection steps 121, 215  
   SP service URL 150, 236  
WS-Trust STS  
   configuring request parameters 280  
   enabling 269  
   IdP connections 303  
   mapping request parameters 286  
   SP connections 281  

X  
X.509 certificates 10  
XASP  
   requester mapping 203  
XML encryption 10  
   about 14  
   exporting keys to metadata 30  
   IdP-to-SP 155  
   SP-to-IdP 242  
XML files, signing 31