

PingFederate[®] 4.4

Administrator's Manual

© 2007 Ping Identity® Corporation. All rights reserved.

Part Number 3007-130
Version 4.4
April, 2007

Ping Identity Corporation
1099 18th Street, Suite 2950
Denver, CO 80202
U.S.A.

Phone: 877.898.2905 (+1 303.468.2882 outside North America)
Fax: 303.468.2909
Web Site: <http://www.pingidentity.com>

Trademarks

PingFederate, Ping Identity, and the Ping Identity logo are trademarks or registered trademarks of Ping Identity Corporation.

All other trademarks or registered trademarks are the property of their respective owners.

Disclaimer

This document is provided for informational purposes only, and the information herein is subject to change without notice. Ping Identity Corporation does not provide any warranties and specifically disclaims any liability in connection with this document.

Contents

	About This Manual	1
	Overview	1
	Intended Audience	2
	Other Documentation	3
	Text Conventions	3
Chapter 1	Introduction	5
	About Federated Identity	5
	PingFederate Capabilities	7
	Possible SSO Scenarios	7
	Critical Features	8
	Multi-protocol Support	8
	Use Case Configuration	8
	Out-of-the-Box Integration	8
	Turnkey Partner Enablement	9
	Enterprise Deployment Architecture	9
	Support	10
	Services	11
Chapter 2	Standards Support	13
	Federation Roles	13
	Terminology	14
	SAML 1.x Profiles	16
	SSO: Browser/POST	16
	SSO: Browser/Artifact	18
	SP-Initiated (“Destination-First”) SSO	19
	SAML 2.0 Profiles	20
	Single Sign-on	20
	SP-Initiated SSO: POST/POST	20
	SP-Initiated SSO: Redirect/POST	22

	SP-Initiated SSO: Artifact/POST	23
	SP-Initiated SSO: POST/Artifact	25
	SP-Initiated SSO: Redirect/Artifact	26
	SP-Initiated SSO: Artifact/Artifact	27
	IdP-Initiated SSO: POST	28
	IdP-Initiated SSO: Artifact	30
	Single Logout	31
	About Session Clean-up	31
	Attribute Query and XASP	31
	IdP Discovery	32
	WS-Federation	32
	Passive Requestor Profile	33
	Account Linking	34
	Transport and Message Security	35
Chapter 3	Key Concepts	37
	Identity Mapping	37
	Account Linking	38
	Linking Permission and “Defederation”	38
	SP Affiliations	38
	Account Mapping	38
	Integration Kits and Adapters	39
	Bundled Adapters	39
	Commercial Adapters	40
	Software Development Kit	40
	About Attributes	40
	Attribute Contracts	41
	Adapter Contracts	41
	Extended Adapter Contract	42
	Data Stores	42
	Attribute Masking	42
	Certificates, SSL, and XML Encryption	43
	Digital Signatures	43
	Message Signing	43
	Signature Validation	44
	Certificate Validation	44
	Digital Signing Policy Coordination	44
	Secure Sockets Layer	45
	SAML SSL/TLS Scenarios	45
	Authentication	46
	Verifying Trusted Certificates	46
	XML Encryption	46

	Federation Planning Checklist	46
	Configuration Data Exchange	49
	IdP to SP	49
	SP to IdP	49
	Mutual Settings Between Parties	50
Chapter 4	Installation	51
	System Requirements	52
	Operating Systems	52
	User Store/Data Store Integration	52
	Browsers	52
	Java Environment	53
	Minimum Hardware Requirements	53
	Installing the JDK	53
	Installing PingFederate	54
	Running PingFederate for the First Time	54
	Deployment Options	56
	Installing PingFederate as a Service	58
	Uninstalling PingFederate	60
	Uninstalling Services	60
Chapter 5	Console Navigation	63
	Using the Main Menu	63
	Navigating the Administrative Console	65
	About Tasks and Steps	65
	Console Buttons	67
Chapter 6	System Settings	69
	Managing Server Settings	69
	Setting Administration Style	70
	Entering System Information	71
	Configuring Notification Options	71
	License Notification Address	72
	Managing Accounts	73
	Choosing Roles and Protocols	74
	Specifying Federation Information	75
	Changing Session Timeout	76
	Saving and Editing Server Settings	77
	Managing Data Stores	77
	Configuring a JDBC Database Connection	79
	Setting Advanced Options	81
	Configuring an LDAP Connection	82
	Configuring a Custom Data Store	84
	Configuring a Custom Data Store Instance	85
	Invoking Adapter Actions	85
	Editing and Saving Data Store	86

	Configuring IdP Discovery	86
	Configuring a Common Domain Service	87
	Configuring a Local Common Domain Server	88
	Editing and Saving the Configuration	88
Chapter 7	System Administration	89
	Starting and Stopping PingFederate	90
	Log File Generation	90
	Administrator Audit Logging	91
	Runtime Transaction Logging	92
	Transaction Logging Modes	93
	Exporting Metadata	94
	Defining Metadata Attribute Contracts	95
	Choosing a Metadata Signing Key	96
	Export XML Encryption Certificates	96
	Completing the Export	97
	Signing XML Files	97
	Using the Configuration Archive	99
	Managing Email Configuration	100
	Account Management	101
	Setting or Resetting Passwords	103
	Changing Passwords	105
	Using Virtual Host Names	105
	Installing a New License Key	106
	Locating Your Server ID	107
	Changing Configuration Parameters	107
	Using Velocity Templates	110
Chapter 8	Security Management	111
	Trusted CAs	111
	SSL Server Certificates	113
	SSL Client Keys & Certificates	115
	Digital Signing and Decryption Keys & Certificates	118
	Application Authentication	120
Chapter 9	Identity Provider Configuration	123
	Application Integration Settings	124
	Configuring IdP Adapters	124
	Selecting an IdP Adapter Type	125
	Configuring an IdP Adapter	126
	Invoking Adapter Actions	126
	Extending an Adapter Contract	127
	Setting Pseudonym Values and Masking	128
	Selecting an Authentication Context	129
	Editing and Saving Adapter Instances	130
	Configuring a Default URL and Error Message	130
	Viewing Application Endpoints	131
	Viewing Protocol Endpoints	131

Configuring SP Connections	132
Accessing Connections	133
Using the Main Menu	133
Using the Connection List Screen	134
Configuration Steps	136
Selecting a Protocol	138
Importing Metadata	138
Importing a Verification Certificate	138
Viewing the Metadata Summary	138
General Information	139
Setting an Assertion Lifetime	141
Choosing SAML Profiles	141
Configuring Web SSO	143
Configuring IdP-Initiated SSO	144
Configuring SP-Initiated SSO	144
Configuring IdP-Initiated SLO	145
Configuring SP-Initiated SLO	146
SSO/SLO Profile Configuration Steps	146
Configuring Identity Mapping	147
Creating an Attribute Contract	149
IdP Adapter Mapping	151
Specifying a Failsafe Attribute Source	168
Mapping Default Attribute Contract Fulfillment	169
Setting Assertion Consumer Service URLs (SAML)	171
Setting a Default Target URL (SAML 1.x)	172
Defining a Service URL (WS-Federation)	173
Specifying SLO Service URLs (SAML 2.0)	174
Choosing Allowable SAML Bindings (SAML 2.0)	175
Setting an Artifact Lifetime (SAML)	176
Specifying Artifact Resolver Locations (SAML 2.0)	177
Configuring Signature Policy	178
Configuring XML Encryption Policy (SAML 2.0)	178
Editing and Saving Web SSO Configurations	180
Configuring the Attribute Query Profile	180
Defining Retrievable Attributes	180
Choosing a Data Store	181
Configuring Data Store Lookup	182
Attribute Mapping Fulfillment	182
Specifying Security Policy	183
Editing and Saving Attribute Query Configurations	184
Configuring Credentials	184
Configuring Back-Channel Authentication	185
Configuring Digital Signature Settings	188
Selecting Signature Verification Certificates	189
Selecting an Encryption Certificate (SAML)	190

	Selecting a Decryption Key (SAML)	191
	Editing and Saving Credential Configurations	192
	Connection Activation and Summary	192
	Defining SP Affiliations	193
	Using the Affiliations List Screen	194
	Importing Affiliation Metadata	194
	Entering Affiliation Information	195
	Managing Affiliation Membership	195
Chapter 10	Service Provider Configuration	197
	Application Integration Settings	198
	Configuring SP Adapters	198
	Creating an Adapter Instance	199
	Configuring an Adapter Instance	201
	Invoking Adapter Actions	202
	Extending an Adapter Contract	203
	Editing and Saving SP Adapter Instances	204
	Mapping URLs to SP Adapter Instances	205
	Configuring Default URLs	207
	Viewing Application Endpoints	207
	Federation Settings	208
	Attribute Requester Mapping	208
	Viewing Protocol Endpoints	210
	Configuring IdP Connections	211
	Accessing Connections	211
	Using the Main Menu	212
	Using the Connection List Screen	213
	Configuration Steps	214
	Selecting a Protocol	216
	Importing Metadata	216
	Importing a Verification Certificate	217
	Viewing the Metadata Summary	217
	General Information	217
	Choosing SAML Profiles	220
	Configuring Web SSO	221
	Configuring IdP-Initiated SSO	222
	Configuring SP-Initiated SSO	223
	Configuring IdP-Initiated SLO	223
	Configuring SP-Initiated SLO	224
	SSO/SLO Profile Configuration Steps	224
	Selecting Identity Mapping	225
	Creating an Attribute Contract	226
	Configuring Adapter Mapping and User Lookup	228
	Specifying SSO Service URLs (SAML)	244
	Specifying a Service URL (WS-Federation)	246
	Specifying SLO Service URLs (SAML 2.0)	246

	Choosing Allowable SAML Bindings (SAML)	248
	Setting an Artifact Lifetime (SAML 2.0)	248
	Specifying Artifact Resolver Locations	249
	Configuring Signature Policy	250
	Configuring XML Encryption Policy (SAML 2.0)	251
	Editing and Saving Web SSO Configurations	253
	Configuring the Attribute Query Profile	253
	Setting the Attribute Authority Service URL	253
	Mapping Attribute Names	254
	Specifying Security Policy	255
	Editing and Saving Attribute Query Configurations	255
	Configuring Credentials	255
	Configuring Back-Channel Authentication	256
	Configuring Digital Signature Settings	259
	Selecting Signature Verification Certificates	260
	Selecting an Encryption Certificate	262
	Selecting a Decryption Key	263
	Editing and Saving Credential Configurations	264
	Connection Activation and Summary	264
Appendix A	Standard Adapter Configuration	267
	Configuring the IdP Standard Adapter	269
	Configuring the SP Standard Adapter	272
	Configuring Advanced Fields	276
Appendix B	LDAP Adapter Configuration	279
	Configuring the IdP LDAP Adapter	281
	Configuring the SP LDAP Adapter	284
Appendix C	Application Endpoints	289
	IdP Endpoints	289
	SP Endpoints	291
Appendix D	Clustering and Failover Deployment	295
	Deployment Scenarios	295
	Basic Clustering	296
	Subclustering	297
	Server Installation	299
	Configuration Deployment	300
	SDK Clustering Extensibility	301
Appendix E	SSO Directory Service	303
	SOAP Request and Response Example	304
	Code Example	305
Appendix F	Using the SafeNet Luna HSM	307
Appendix G	Troubleshooting	311
	Data Stores	312
	Installation	313
	Protocol	313
	Server	313

Glossary	315
List of Acronyms	319

About This Manual

This manual provides information about using Ping Identity's PingFederate to deploy a federated identity management solution based on the latest security and e-business standards.

Overview

The manual consists of:

- [Chapter 1, "Introduction"](#) — A high-level view of identity federation and PingFederate features.
- [Chapter 2, "Standards Support"](#) — An overview of industry standards that PingFederate supports, include the Security Assertion Markup Language (SAML) and WS-Federation.
- [Chapter 3, "Key Concepts"](#) — A discussion of central concepts needed to further understand identity federation and PingFederate.
- [Chapter 4, "Installation"](#) — How to install PingFederate and run the administrative console for the first time.
- [Chapter 5, "Console Navigation"](#) — A tutorial on using the administrative console and configuration screens.
- [Chapter 7, "System Administration"](#) — Information about maintaining the PingFederate server and deployment, using log files, updating license information and managing administrative users.
- [Chapter 6, "System Settings"](#) — How to configure your local PingFederate server to handle identity federation transactions.
- [Chapter 8, "Security Management"](#) — Information about importing, exporting, and maintaining digital signature and SSL certificates in PingFederate.

- [Chapter 10, “Service Provider Configuration”](#) — How to configure PingFederate to establish connections from a Service Provider to an Identity Provider.
- [Chapter 9, “Identity Provider Configuration”](#) — How to configure PingFederate to establish connections from an Identity Provider to a Service Provider.
- [Appendix A, “Standard Adapter Configuration”](#) — How to configure PingFederate to use the packaged Standard Adapter for interfacing your Web applications.
- [Appendix B, “LDAP Adapter Configuration”](#) — How to configure the packaged LDAP SP Authentication Adapter for providing identity management and interfaces to Web applications.
- [Appendix C, “Application Endpoints”](#) — Detailed information about using PingFederate connection endpoints for Web single sign-on and single logout.
- [Appendix D, “Clustering and Failover Deployment”](#) — How to deploy PingFederate for redundancy and high-availability.
- [Appendix E, “SSO Directory Service”](#) — Web-service interface information for developers using the PingFederate Software Development Kit.
- [Appendix F, “Using the SafeNet Luna HSM”](#) — How to install and configure PingFederate with the Luna SA Hardware Security Module as one step in the process to become Federal Information Processing Standard (FIPS) 140-2 complaint.
- [Appendix G, “Troubleshooting”](#) — How to resolve potential configuration and deployment issues.
- [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 101).

Other Documentation

Sample Application Quick Start Guide The PingFederate *Sample Application Quick Start Guide*, located in the `/quickstart/docs` directory, provides procedures for setting up a deployment of PingFederate on a Windows server to run with sample 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.

Web Resources Ping Identity continuously updates its Web site with general and technical information in the form of White Papers, FAQs, Tech Notes, and other resources—at www.pingidentity.com.



Tip: PingFederate also provides context-sensitive online help. Click **Help** in the upper-right portion of the administrative console for immediate guidance, along with links to related information.

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](mailto:support@pingidentity.com) (support@pingidentity.com) if you encounter a problem.

Text Conventions

This document uses text conventions identified below.

Table 1: Text Convention Definitions

Convention	Description
Fixed Width	Indicates text that must be typed exactly as shown in the instructions. Also used to represent program code, file names, and directory paths.
Blue text	Used in online documents to indicate hypertext links.
<i>Italic</i>	Used for emphasis and to identify document titles and new terms.
► [text]	Used for procedures where only one step is required.
Sans serif	Identifies GUI text as shown on a screen. Example: Print Document dialog
Sans serif bold	Identifies menu items, navigational links, or buttons. For example: Click Save .

Introduction

Ping Identity's PingFederate is the industry-leading federated identity server for enabling single sign-on (SSO) to online services for employees, customers, and business partners. No other federated identity solution is as easy to install, integrate, and scale. PingFederate reduces complexity, cost, and time-to-production through effortless configuration tools and the broadest range of turnkey integration kits.

About Federated Identity

Federated identity management (or “identity federation”) enables enterprises to securely link and exchange identity information across partner, supplier, and customer organizations (see [Figure 1](#) on page 6). Federation takes an open, standards-based approach that eliminates the cost overruns, security loopholes, and user inconvenience caused by rigid, proprietary, “siloed” application architectures.

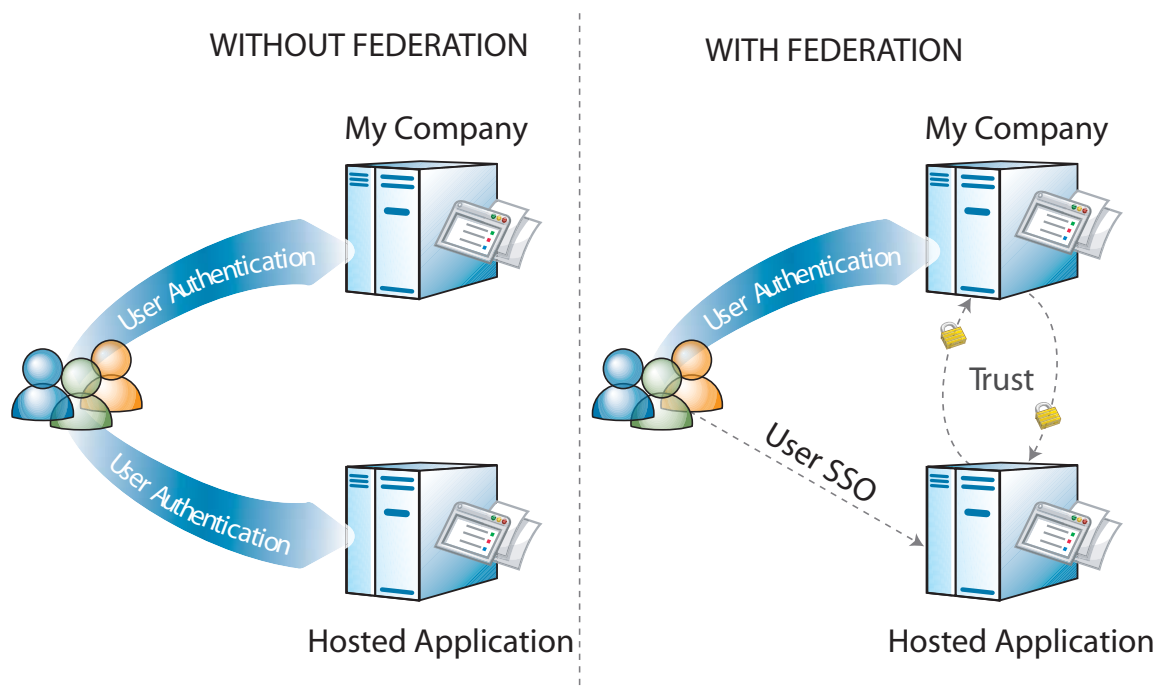


Figure 1: Federated Single Sign-on

This cross-domain identity management solution provides numerous benefits, ranging from enhanced customer relations and service to reduced cost and greater security and accountability:

Improve Customer Retention/Service Identity federation provides organizations with the ability to streamline the delivery of services. As a service provider, your customers' employees can use their own network user IDs and passwords to access your online services without having to create, manage, and remember new passwords. The burden on end-users is lightened and their satisfaction increased. The barrier to using your services is lowered, and your cost of managing user accounts is dramatically reduced.

Optimize Business Processes Companies that manage user accounts for a large number of external users (customers, suppliers, and business partners) can benefit dramatically from deploying an identity federation. As organizations optimize business processes by collaborating more closely with their customers, suppliers, and business partners, the concurrent increase in the identity management costs can skyrocket. These costs can be reduced dramatically by enabling external users' identities to be shared across security domains as part of a collaborative business system using identity federation.

Simplify Collaboration Across Business Units Organizations that consist of multiple operating units or that have grown through acquiring other businesses frequently have entire information systems isolated from other business units within the same organization. Creating an identity federation that spans multiple business units within an organization provides the units with the autonomy to manage their own information systems and users while simultaneously allowing users to gain access to shared systems and resources.

Reduce the Burden of Regulatory Compliance By using an identity federation to create a central access gateway for external users, organizations are also able to more easily comply with corporate governance regulations that require organizations to monitor and manage access to sensitive business information.

Additionally, identity federations can help organizations meet various national privacy regulations by controlling the amount of personal information that is shared among organizations and across national borders.

Increase Information Security Deploying an identity federation reduces the number of passwords that a user has to recall and thereby reduces an organization's exposure to unauthorized access to its sensitive business information.

Reduce IT Costs Due to the overall reduction in the number of passwords and user accounts that each user is required to remember and manage, it is possible for an organization that deploys an identity federation to reduce its user support costs by 80% to 85% (Ant Allan, Gartner Group, "Password Management, Single Sign-On, and Authentication Management Infrastructure Products: Perspective", January 7, 2003).

PingFederate Capabilities

PingFederate's lightweight, stand-alone architecture means you can receive the benefits of standards-based Web SSO without the cost and complexity of a complete identity management (IdM) system. The PingFederate server integrates and coexists with home-grown and commercial identity management deployments without requiring extensive upgrades to an entrenched IdM system.

Possible SSO Scenarios

There are multiple scenarios for enabling single sign-on with PingFederate. For example:

- An organization can enable its customers and business partners to sign on to its online services by deploying PingFederate to act as a federation "hub."
- An organization can enable its employees to sign on to another organization's online services by deploying PingFederate to act as a federation "spoke."
- An organization that has distributed implementations of business applications and back-office systems can enable employees to sign on to all of these systems with a single user ID and password by deploying PingFederate as an internal single sign-on "hub."
- Most organizations that deploy PingFederate in one of the foregoing scenarios quickly identify additional opportunities to realize the benefits associated with the other scenarios. They then choose to extend their deployments into a blended scenario where PingFederate is used *both* as a hub and a spoke.

All of these scenarios are achievable with a single PingFederate implementation.

Critical Features

- Multi-protocol support, including SAML 1.x, 2.0 and WS-Federation, for best-of-breed functionality dedicated to federated SSO, Single logout (SLO), and attribute exchange
- Use-case-driven configuration guides administrators step-by-step through system setup and federation connections
- Out-of-the-box integration simplifies integration with existing applications and minimizes impact on existing infrastructure
- Turnkey partner enablement streamlines partner configuration and connection setup for rapid deployment
- Enterprise-deployment architecture provides centralized and scalable federation management

Multi-protocol Support

PingFederate 3 introduced a new use-case-driven configuration paradigm that radically simplified and accelerated SAML 2.0 configuration. PingFederate 4 extends this innovation to support three more standard federation protocols: SAML 1.0, SAML 1.1, and WS-Federation (see [“Standards Support”](#) on page 13).

Use Case Configuration

By providing a single configuration paradigm supporting four different protocols, PingFederate 4 reduces complexity and learning curves in multiple-protocol federation deployments. Furthermore, the administrative console minimizes the potential for errors by asking the administrator only for configuration parameters that are applicable to the protocols they have indicated they need to support.

Out-of-the-Box Integration

PingFederate is the only federated identity server with a suite of Integration Kits to complete the first- and last-mile integration of sessions with your existing applications. PingFederate Integration Kits are available for download from the Ping Identity Web site, take only minutes to install, and are configured from within the PingFederate GUI administrative console.

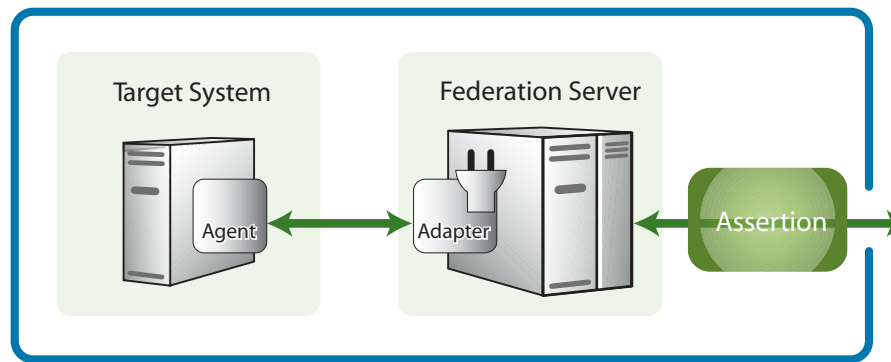


Figure 2: Integration Kit Architecture

Integration Kits enable rapid session integration with both existing authentication services and target applications. In addition, PingFederate includes a Software Development Kit for creating custom integrations.

Visit www.pingidentity.com for the latest information on availability of our Integration Kits.

Turnkey Partner Enablement

Many companies consider adding and configuring new federation partners to be one of the most challenging aspects of federation. Federating with partners often requires education, metadata negotiation, and agreements on liability shift before the technology-related issues can even be addressed. Some companies struggle for weeks with overly complex toolkits or products to establish a reliable federation connection with just a single partner.

In addition to the technical challenges that organizations face when establishing identity federations, a number of procedural hurdles also exist. In order to establish the trust relationships required for each partner to join an identity federation, a number of operational and legal agreements must first be put in place.

Ping Identity Corporation provides a comprehensive suite of turnkey partner programs to streamline and expedite the operational and legal aspects of forming an identity federation. In addition, Ping Identity Client Services is available to provide expert consultation on all types of issues (technical, operational, and legal) to expedite the deployment of the identity federation.

Enterprise Deployment Architecture

PingFederate is the only federation server that enables you to federate applications residing in multiple security domains using different protocols, effectively allowing you to manage all of your partner trust relationships and connections from a single location.

PingFederate fully supports Burton Group's recommended architecture for a stand-alone server, thus enabling efficient, platform-level scalability for all your federation initiatives.

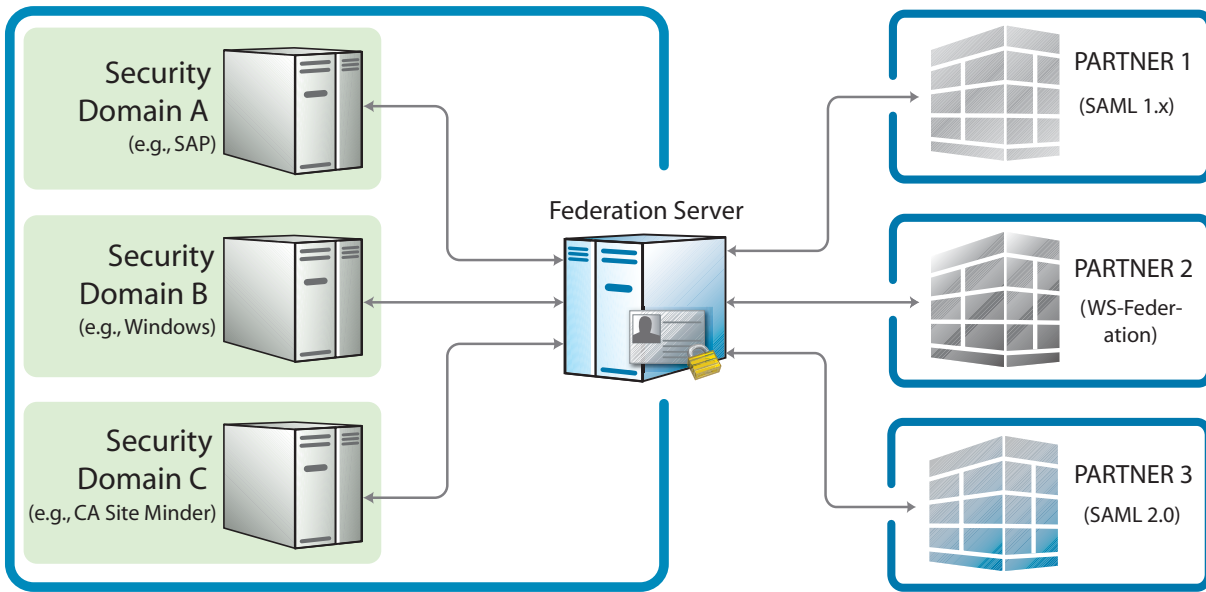


Figure 3: Multiple-security-domain, cross-protocol federation

PingFederate's unique capacity to connect to applications residing in multiple security domains using different protocols, simplifies enterprise deployment and reduces redundant deployment and administration.

With PingFederate's Enterprise Deployment Architecture, all protocol definitions, public key infrastructure (PKI) keys, policies, [attribute contracts](#), profiles, etc. are managed in a single location, eliminating the need to maintain redundant copies of these configurations and trust relationships. Furthermore, when new protocols, profiles, or use cases need to be added, you only have to configure them once to make them available to your entire organization.

PingFederate's enterprise deployment architecture also improves security by creating a single “doorway” in your perimeter through which all identity information must travel. Using PingFederate, all of your internal users who single sign-on to external applications will exit through this doorway, while all external users who single sign-on to your internal systems will enter through the same doorway.

The single-doorway approach also provides 100 percent visibility to all federation activities. The extensive auditing and logging capabilities of PingFederate enable you to satisfy all of your logging-related compliance and service-level requirements from a single location, as opposed to having to acquire and consolidate disparate logs from throughout your organization.

Support

Ping Identity offers a variety of support services, affording customers the option of designing their own individualized service configuration for PingFederate.

There are three “families” of support services, as described below. Each can be customized with options to extend the hours of support coverage, the type of services required, and the response time desired.

PingEnable This suite of services is designed to assist customers with the rapid deployment of PingFederate, offering fixed-price and predefined deliverable-based quick-start, training, and implementation services.

PingConsulting Custom professional services are available to assist our customers with specialized implementation, training, or development above and beyond PingEnable services.

PingProduction Maintenance & Support (M&S) services are provided via three distinct offerings (each with configurable options and extensible features) defined using the International Olympics nomenclature of Bronze, Silver, and Gold service plans.

If you have an existing support contract with Ping Identity, please feel free to send any inquiries to support@pingidentity.com, or you may contact the phone number listed in your “Welcome to PingFederate Support” email.

If you do not have an existing support contract with Ping Identity and you would like to discuss your service options, please contact sales@pingidentity.com or call toll-free 877.898.2905 (+1 303.468.2882 outside North America).

Services

Ping Identity is prepared to assist at any point in the federation process—including team briefings, technology and business planning, implementation, and rollout. Standard offerings are available for each project stage and can be tailored to meet your specific needs.

Orientation

Executive briefings on federation technologies and implementation strategies, including:

- A review the benefits of federated identity and return on investment
- Learning about deployment options, common use cases, integration strategies and ongoing administrative requirements

Planning

- Architecture and integration strategies
- Consultation on federation legal agreements
- Workshops for evaluating liability

Implementation

- Federation quick-start
- Implementing a proof of concept or prototype

- Migrating from a proprietary implementation to a standards-based, scalable solution using PingFederate

Support and Partner Enablement

Federation integration, technology migration, and support, including:

- Migrating to PingFederate from proprietary protocols
- Assisting multiple business partners in joining your federation
- Accessing on-demand federation expertise

For further information contact sales@pingidentity.com or call toll-free 877.898.2905 (+1 303.468.2882 outside North America).

Standards Support

PingFederate provides flexible, integrated support for all versions of the Security Assertion Markup Language (SAML) protocol, from 1.0 through 2.0. In addition, PingFederate supports the WS-Federation browser-based, “passive” protocol using SAML assertions as SSO-enabling security tokens.

This chapter describes:

- [“Federation Roles”](#) on page 13
- [“SAML 1.x Profiles”](#) on page 16
- [“SAML 2.0 Profiles”](#) on page 20
- [“WS-Federation”](#) on page 32
- [“Account Linking”](#) on page 34
- [“Transport and Message Security”](#) on page 35

Federation Roles

The most recent sets of standards, SAML 2.0 and WS-Federation, define two roles in an identity federation partnership: an Identity Provider (IdP) and a Service Provider (SP).



Note: Earlier SAML 1.x specifications used the terms Asserting Party (for IdP) and Relying Party (for SP). For consistency and clarity, however, PingFederate adopts the later terms IdP and SP across all specifications.

A third role, defined in the specifications and available in PingFederate, is that of an IdP Discovery provider.

Identity Provider

An IdP, also called the “SAML authority,” is a system entity that authenticates a user, or “SAML subject,” and transmits referential identity information based on that authentication.



Note: The SAML subject may be a person, a Web application, or a Web server. Since the subject is often a person, the term “user” is generally employed throughout this manual.

Service Provider

An SP is the consumer of identity information provided by the IdP. Based on trust, technical agreements, and verification of adherence to protocols, SP applications and systems determine whether (or how) to use information contained in a SAML assertion.

IdP Discovery Provider

This role provides an IdP look-up service that can be incorporated into the implementation of either an IdP or an SP, or it can be employed as a stand-alone server (see “[IdP Discovery](#)” on page 32).

Terminology

The SAML specifications provides a system of building blocks and support components for achieving secure data exchange in an identity federation. These include:

- [Assertions](#)
- [Bindings](#)
- [Profiles](#)
- [Metadata](#) (SAML 2.0)
- [Authentication Context](#) (SAML 2.0)

Assertions

Assertions are XML documents sent from an IdP to an SP. Each assertion contains identifying information about a user who has initiated an SSO request.

Bindings

A SAML binding describes the way messages are exchanged using transport protocols. PingFederate supports the following bindings:

- **HTTP POST** – Describes how SAML messages are transported in HTML form-control content, which uses a base-64 format.
- **HTTP Artifact** – Describes how to use an artifact to represent a SAML message. The artifact can be transported via an HTML form control or a query string in the URL.
- **HTTP Redirect (SAML 2.0)** – Describes how SAML messages are transported using HTTP 302 status-code response messages.
- **SOAP (SAML 2.0)** – Describes how SAML messages are to be transferred across the [back channel](#).

Profiles

Profiles describe processes and message flows combining assertions, request/response message specifications, and bindings to achieve a specific desired functionality or use case. Because profiles define the application of the specifications and therefore play a large part in PingFederate, most of the rest of this chapter is devoted to them, starting with [“SAML 1.x Profiles”](#) on page 16.

Metadata

SAML 2.0 defines an XML schema to standardize metadata to facilitate the exchange of configuration information among federation partners. This information includes, for example, profile and binding support, connection endpoints, and certificate information. (See [“Exporting Metadata”](#) on page 94.)

Whether you are exporting or importing a metadata file, PingFederate supports the use of XML digital signatures to ensure the integrity of the data (see [“Signing XML Files”](#) on page 97).

Authentication Context

Before allowing access to a protected resource, an SP may want background information surrounding how the user was originally authenticated by the IdP, in addition to the assertion itself. The SP may use this information for an access control decision or to provide an audit trail for regulatory or security policy compliance.

The exact content of the authentication context is left up to the federation partners to interpret and implement. Each IdP will use a different set of authentication technologies, follow different processes, and be bound by different legal obligations regarding authentication.

The SAML specification provides an XML schema whereby partners can create authentication context declarations. Partners may choose to implement a set of classes provided by the specification to help categorize and simplify context interpretation. Partners may choose to reference a SAML 2.0 URN that identifies each of these classes. However, it is up to partners to decide if

additional authentication context is required and if these classes supply an adequate description.

Several PingFederate integration kits provide methods that can be used by the developer to insert authentication context from external IdP applications into the assertion (see [“Integration Kits and Adapters”](#) on page 39). Conversely, the SP developer can call methods for extracting authentication context from an assertion. It is up to the SP developer and application to create access control or other processing based on the context.

Check the *User Guide* for your integration kit to see if this feature is supported. The Standard Adapter packaged with PingFederate supports this feature.

For more information on configuring authentication context for an adapter instance, see [“Configuring IdP Adapters”](#) on page 124.

SAML 1.x Profiles

SAML 1.0 and 1.1 profiles provide for SSO, initiated by an IdP, using either the POST or Artifact Bindings.

In addition, the specifications provide for a non-normative SP-initiated scenario (called “destination-first”), which allows Web developers to create applications that enable a user to initiate SSO from the SP site.

SSO: Browser/POST

In this scenario, a user is logged on to the IdP and attempts to access a resource on a remote SP server. The SAML assertion is transported to the SP via HTTP POST.

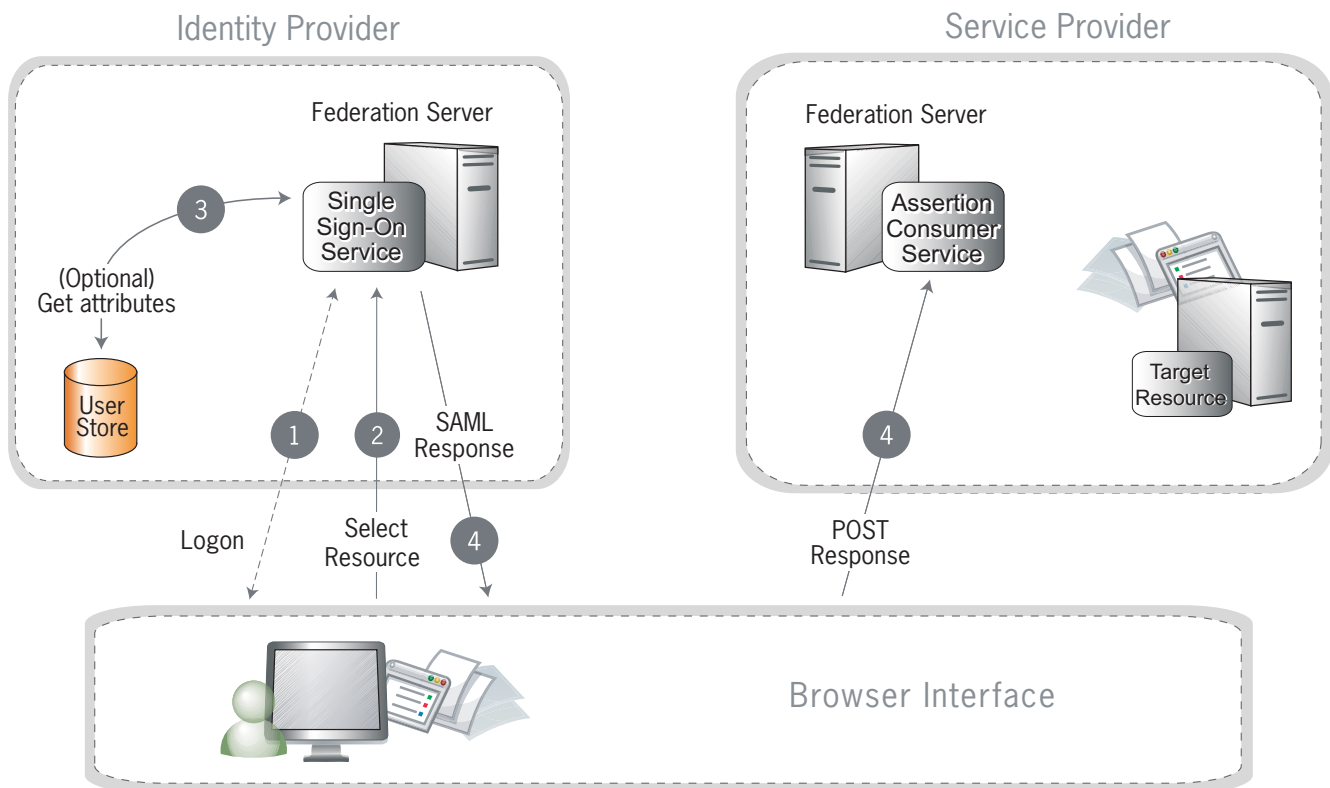


Figure 4: Browser/POST Profile

Processing Steps:

1. A user has logged on to the IdP.
2. The user requests access to a protected SP resource. The user is not logged on to the SP site.
3. Optionally, the IdP retrieves attributes from the user data source.
4. The IdP's SSO service returns an HTML form to the browser with a SAML response containing the authentication assertion and any additional attributes. The browser automatically posts the HTML form back to the SP.



Note: SAML specifications require that POST responses be digitally signed.

5. (Not shown) If the signature and assertion are valid, the SP establishes a session for the user and redirects the browser to the target resource.

SSO: Browser/Artifact

In this scenario, the IdP sends a SAML artifact to the SP via either HTTP POST or a redirect (shown in diagram). The SP uses the artifact to obtain the associated SAML response from the IdP.

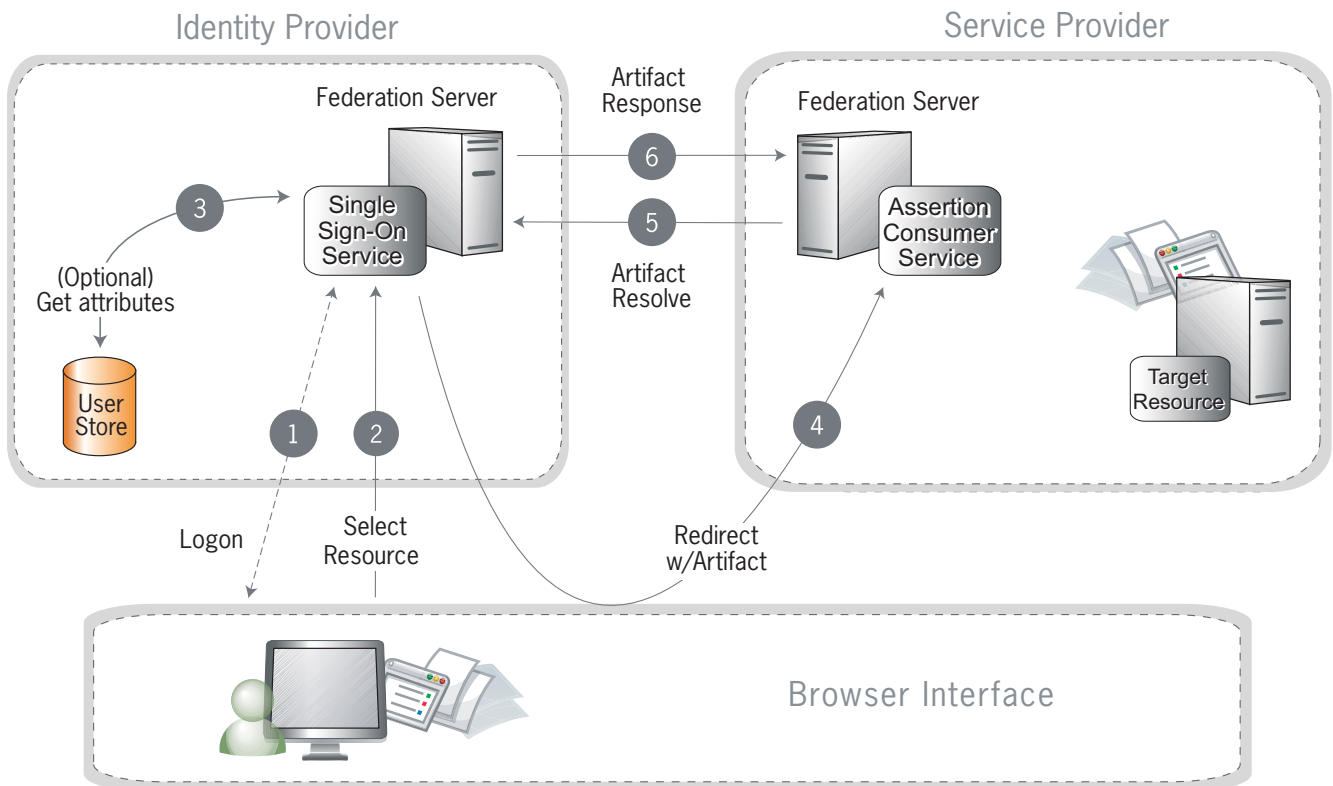


Figure 5: SSO: Browser/Artifact Profile

Processing Steps:

1. A user is logged on to the IdP.
2. The user requests access to a protected SP resource. The user is not logged on to the SP site.
3. Optionally, the IdP retrieves attributes from the user data store.
4. The IdP federation server generates an assertion, creates an artifact, and sends an HTTP redirect containing the artifact through the browser to the SP's Assertion Consumer Service (ACS).
5. The ACS extracts the Source ID from the SAML artifact and sends an artifact-resolve message to the identity federation server's Artifact Resolution Service (ARS).
6. The ARS sends a SAML artifact response message containing the previously generated assertion.
7. (Not shown) If a valid assertion is received, the SP establishes a session and redirects the browser to the target resource.

SP-Initiated (“Destination-First”) SSO

In an SP-initiated (a.k.a. “destination-first”) transaction the user is connected to an SP site and attempts to access a protected resource in the SP domain. The user might have an account at the SP site but according to federation agreement, authentication is managed by the IdP. The SP sends an authentication request to the IdP.

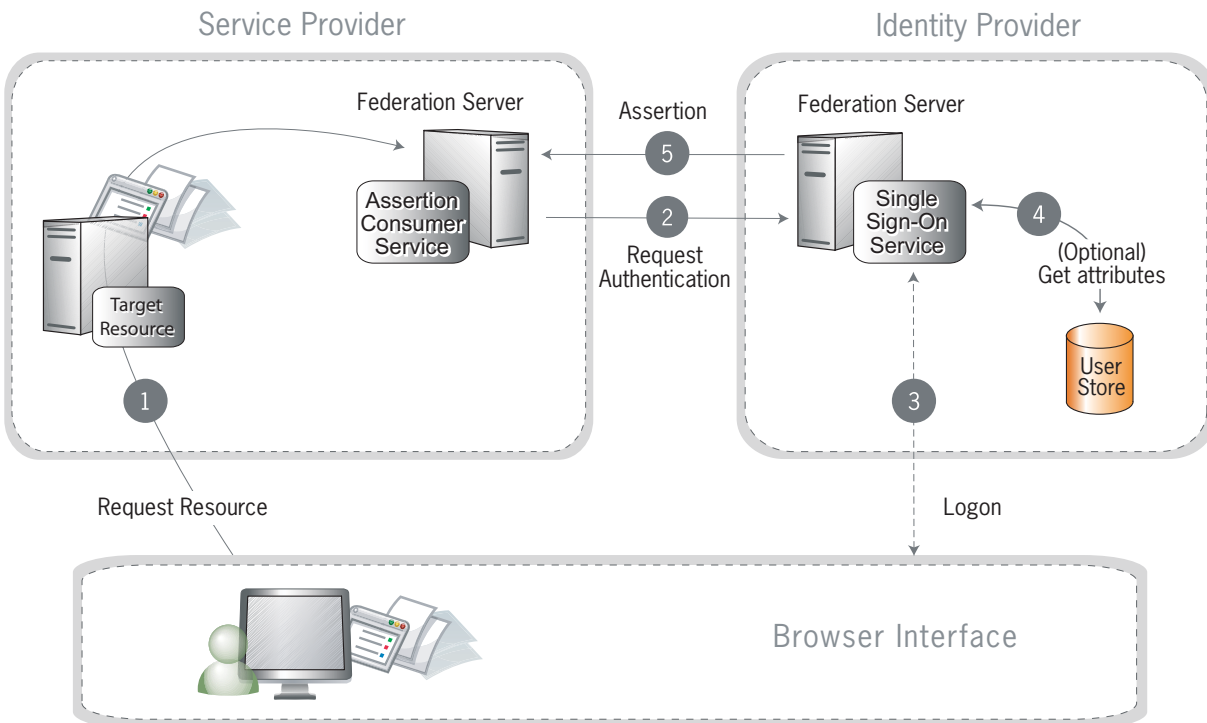


Figure 6: SP-Initiated SSO

Processing Steps:

1. The user requests access to a protected SP resource. The request is redirected to the federation server to handle authentication.
2. The federation server sends a SAML request for authentication to the IdP's SSO service (also called Intersite Transfer Service).
3. If the user is not already logged on to the IdP site or if re-authentication is required, the IdP asks for credentials (e.g., ID and password) and the user logs on.
4. Additional information about the user may be retrieved from the user data store for inclusion in the SAML response. (These attributes are predetermined as part of the federation agreement between the IdP and the SP—see “About Attributes” on page 40.)
5. The IdP's SSO service returns a SAML response containing the authentication assertion and any additional attributes to the SP.
6. (Not shown) If the assertion is valid, the SP establishes a session for the user and redirects the browser to the target resource.

SAML 2.0 Profiles

PingFederate supports these major profiles defined under the SAML 2.0 standard:

- [Single Sign-on](#)
- [Single Logout](#)
- [Attribute Query and XASP](#)
- [IdP Discovery](#)

Single Sign-on

SAML 2.0 substantially increases the number of possible SSO profile variations by fully enabling SP-initiated transactions. When SP- and IdP-initiated protocols are paired with transport [binding](#) specifications, the combinations result in eight practical SSO scenarios:

- [SP-Initiated SSO: POST/POST](#)
- [SP-Initiated SSO: Redirect/POST](#)
- [SP-Initiated SSO: Artifact/POST](#)
- [SP-Initiated SSO: POST/Artifact](#)
- [SP-Initiated SSO: Redirect/Artifact](#)
- [SP-Initiated SSO: Artifact/Artifact](#)
- [IdP-Initiated SSO: POST](#)
- [IdP-Initiated SSO: Artifact](#)

SP-Initiated SSO: POST/POST

In this scenario a user attempts to access a protected resource directly on an SP Web site without being logged on. The user does not have an account on the SP site, but does have a federated account managed by a third-party IdP. The SP sends an authentication request to the IdP. Both the request and the returned SAML assertion are sent through the user's browser via HTTP POST.

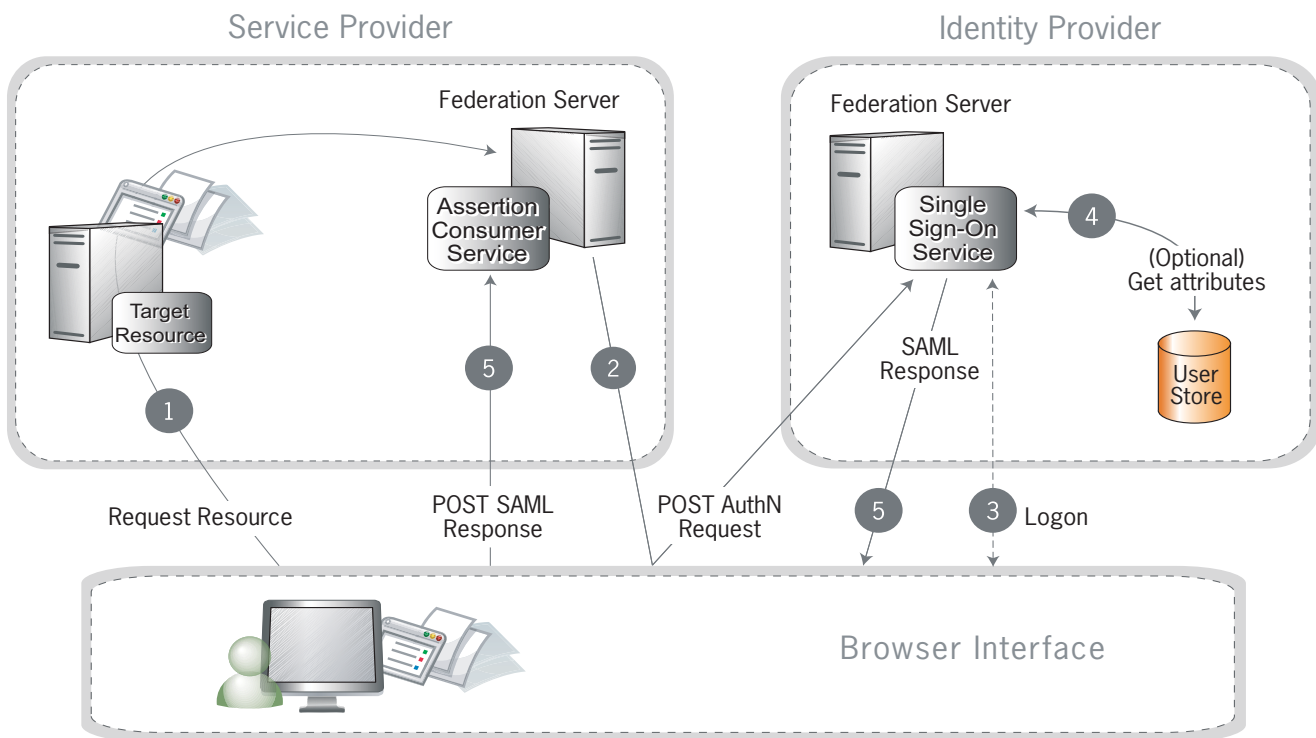


Figure 7: SP-Initiated SSO: POST/POST

Processing Steps:

1. The user requests access to a protected SP resource. The request is redirected to the federation server to handle authentication.
2. The federation server sends an HTML form back to the browser with a SAML request for authentication from the IdP. The HTML form is automatically posted to the IdP's SSO service.
3. If the user is not already logged on to the IdP site or if re-authentication is required, the IdP asks for credentials (e.g., ID and password) and the user logs on.
4. Additional information about the user may be retrieved from the user data store for inclusion in the SAML response. (These attributes are predetermined as part of the federation agreement between the IdP and the SP—see [“About Attributes”](#) on page 40.)
5. The IdP's SSO service returns an HTML form to the browser with a SAML response containing the authentication assertion and any additional attributes. The browser automatically posts the HTML form back to the SP.



Note: SAML specifications require that POST responses be digitally signed.

6. (Not shown) If the signature and assertion are valid, the SP establishes a session for the user and redirects the browser to the target resource.

SP-Initiated SSO: Redirect/POST

In this scenario, the SP sends an HTTP redirect message to the IdP containing an authentication request. The IdP returns a SAML response with an assertion to the SP via HTTP POST.

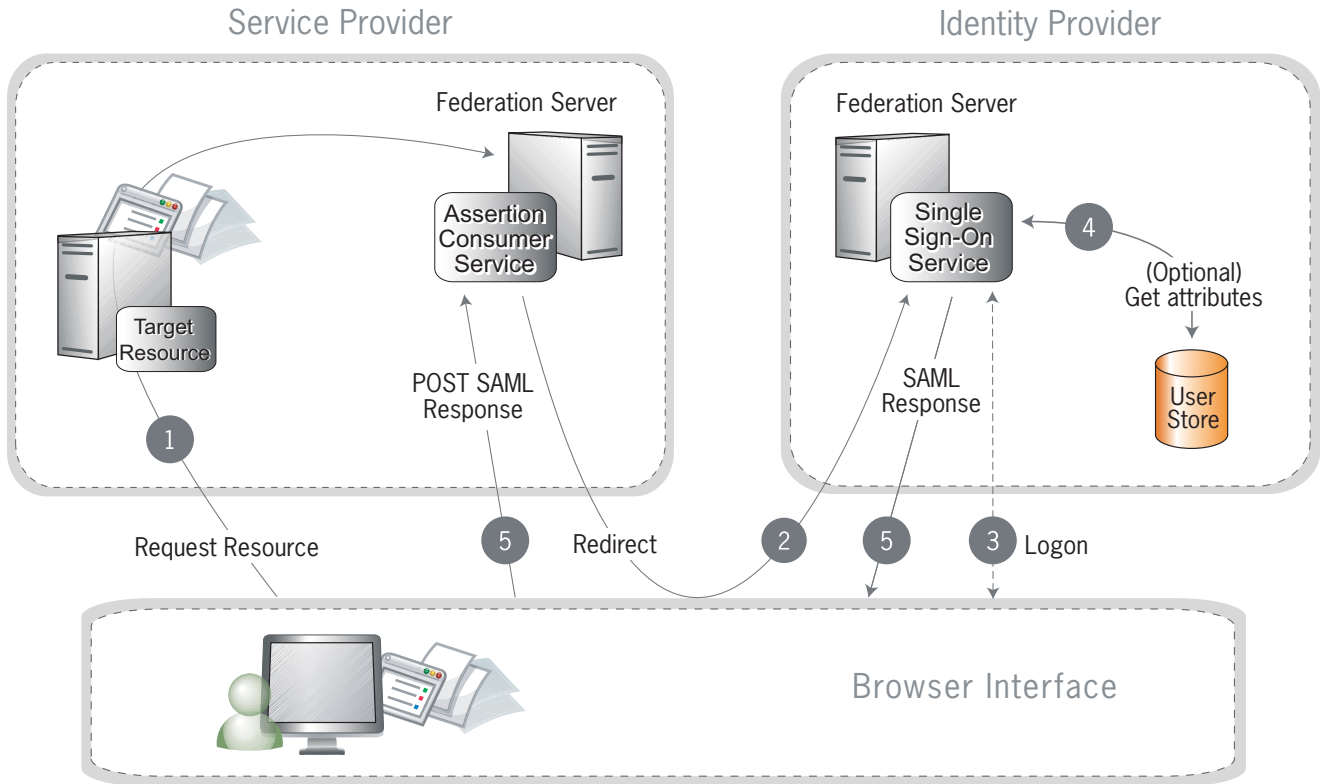


Figure 8: SP-Initiated SSO: Redirect/POST

Processing Steps:

1. A user requests access to a protected SP resource. The user is not logged on to the site. The request is redirected to the federation server to handle authentication.
2. The SP returns an HTTP redirect (code 302 or 303) containing a SAML request for authorization through the user's browser to the IdP's SSO service.
3. If the user is not already logged on to the IdP site or if re-authentication is required, the IdP asks for credentials (e.g., ID and password) and the user logs on.
4. Additional information about the user may be retrieved from the user data store for inclusion in the SAML response. (These attributes are predetermined as part of the federation agreement between the IdP and the SP—see [“About Attributes”](#) on page 40.)

- The IdP's SSO service returns an HTML form to the browser with a SAML response containing the authentication assertion and any additional attributes. The browser automatically posts the HTML form back to the SP.



Note: SAML specifications require that POST responses be digitally signed.

- (Not shown) If the signature and assertion are valid, the SP establishes a session for the user and redirects the browser to the target resource.

SP-Initiated SSO: Artifact/POST

In this scenario, the SP sends a SAML [artifact](#) to the IdP via an HTTP redirect. The IdP uses the artifact to obtain an authentication request from the SP's SAML artifact resolution service. The IdP returns a SAML response to the SP via HTTP POST.

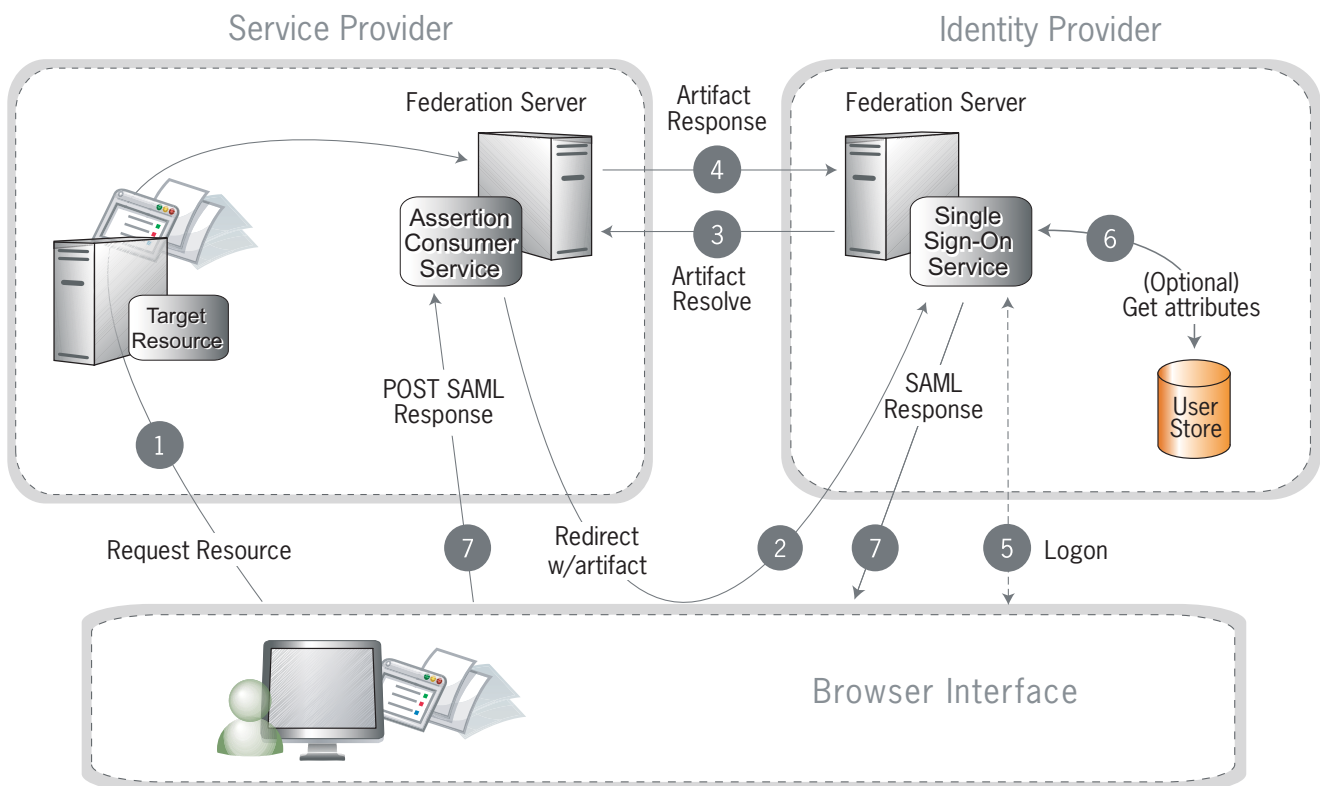


Figure 9: SP-Initiated SSO: Artifact/POST

Processing Steps:

- A user requests access to a protected SP resource. The user is not logged on to the site. The request is redirected to the federation server to handle authentication.

2. The SP generates an authentication request and creates an artifact. The SP sends an HTTP redirect containing the artifact through the user's browser to the IdP's SSO service.



Note: The artifact contains the source ID of the SP's artifact resolution service and a reference to the authentication.

3. The SSO service extracts a source ID from the SAML artifact and sends a SAML artifact-resolve message over [SOAP](#) containing the artifact to the SP's [Artifact Resolution Service](#) (ARS).



Note: The SP and IdP's source IDs and remote artifact resolution services are mapped according to the federation agreement made prior to this action.

4. The SP's ARS returns a SAML message containing the previously generated authentication request.
5. If the user is not already logged on to the IdP site or if re-authentication is required, the IdP asks for credentials (e.g., ID and password) and the user logs on.
6. Additional information about the user may be retrieved from the user data store for inclusion in the SAML response. (These attributes are predetermined as part of the federation agreement between the IdP and the SP—see [“About Attributes”](#) on page 40.)
7. The IdP's SSO service returns an HTML form to the browser with a SAML response containing the authentication assertion and any additional attributes. The browser automatically posts the HTML form back to the SP.



Note: SAML specifications require that POST responses be digitally signed.

8. (Not shown) If the signature and assertion are valid, the SP establishes a session for the user and redirects the browser to the target resource.

SP-Initiated SSO: POST/Artifact

In this scenario, the SP sends an authentication request to the IdP via HTTP POST. The returned SAML assertion is redirected through the user's browser. The response contains a SAML [artifact](#).

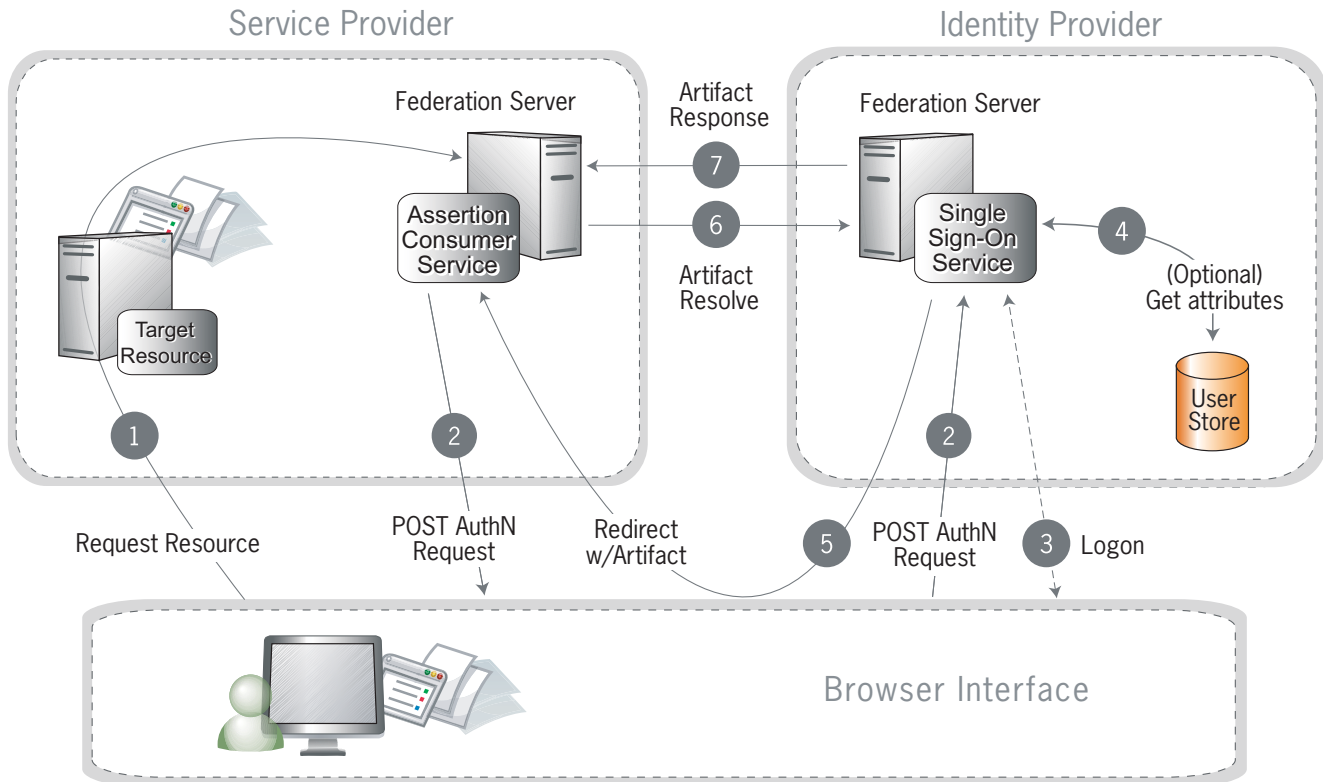


Figure 10: SP-Initiated SSO: POST/Artifact

Processing Steps:

1. A user requests access to a protected SP resource. The user is not logged on to the site. The request is redirected to the federation server to handle authentication.
2. The federation server sends an HTML form back to the browser with a SAML request for authentication from the IdP. The HTML form is automatically posted to the IdP's SSO service.
3. If the user is not already logged on to the IdP site or if re-authentication is required, the IdP asks for credentials (e.g., ID and password) and the user logs on.
4. Additional information about the user may be retrieved from the user data store for inclusion in the SAML response. (These attributes are predetermined as part of the federation agreement between the IdP and the SP—see [“About Attributes”](#) on page 40.)
5. The IdP federation server generates an assertion, creates an artifact, and sends an HTTP redirect containing the artifact through the browser to the SP's [Assertion Consumer Service \(ACS\)](#).

6. The ACS extracts the source ID from the SAML artifact and sends an artifact-resolve message to the federation server's [Artifact Resolution Service](#) (ARS).
7. The ARS sends a SAML artifact response message containing the previously generated assertion.
8. (Not shown) If a valid assertion is received, a session is established on the SP and the browser is redirected to the target resource.

SP-Initiated SSO: Redirect/Artifact

In this scenario, the SP sends an HTTP redirect message to the IdP containing a request for authentication. The IdP returns an [artifact](#) via HTTP redirect. The SP uses the artifact to obtain the SAML response.

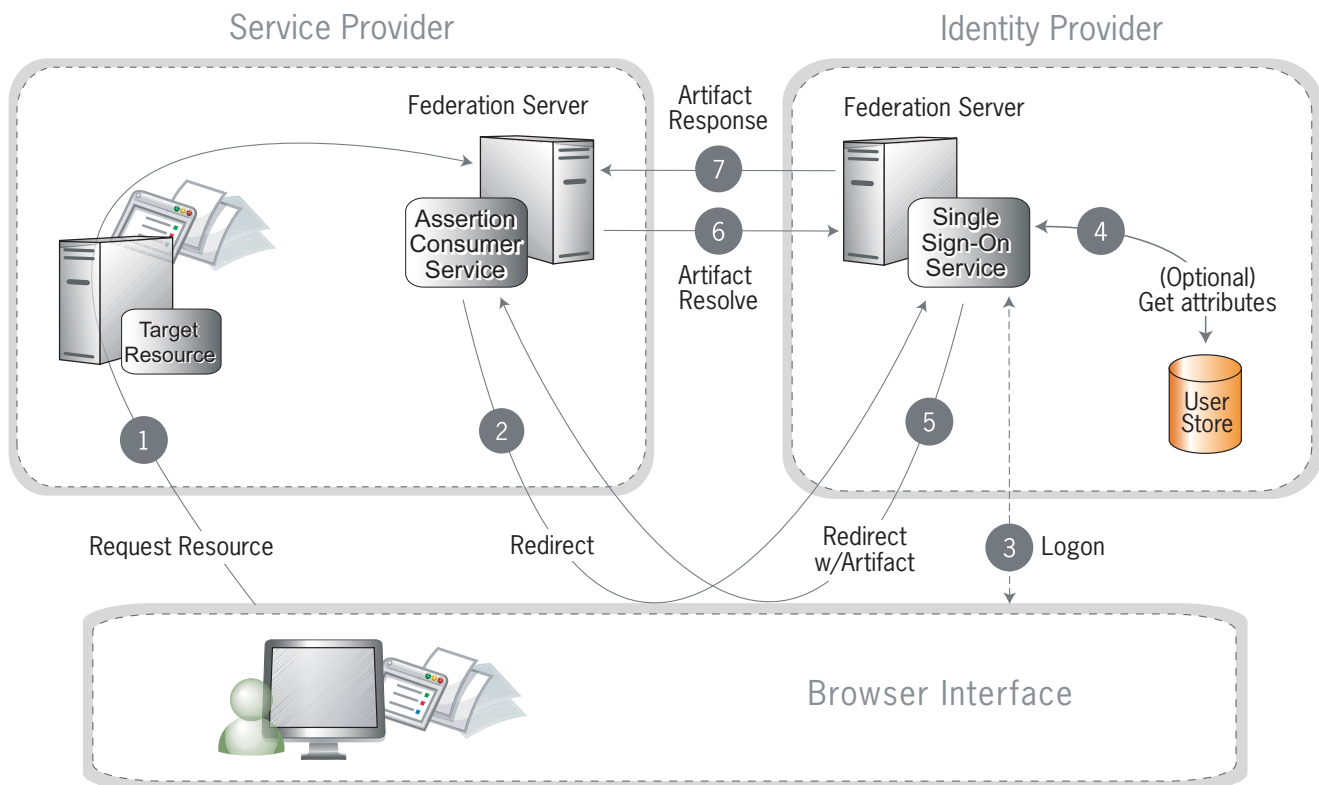


Figure 11: SP-Initiated SSO: Redirect/Artifact

Processing Steps:

1. A user requests access to a protected SP resource. The user is not logged on to the site. The request is redirected to the federation server to handle authentication.
2. The SP returns an HTTP redirect (code 302 or 303) containing a SAML request for authorization through the user's browser to the IdP's SSO service.
3. If the user is not already logged on to the IdP site or if re-authentication is required, the IdP asks for credentials (e.g., ID and password) and the user logs on.

4. Additional information about the user may be retrieved from the user data store for inclusion in the SAML response. (These attributes are predetermined as part of the federation agreement between the IdP and the SP—see “[About Attributes](#)” on page 40.)
5. The IdP federation server generates an assertion, creates an artifact, and sends an HTTP redirect containing the artifact through the browser to the SP’s Assertion Consumer Service (ACS).
6. The ACS extracts the Source ID from the SAML artifact and sends an artifact-resolve message to the identity federation server’s Artifact Resolution Service (ARS).
7. The ARS sends a SAML artifact response message containing the previously generated assertion.
8. (Not shown) If a valid assertion is received, the SP establishes a session and redirects the browser to the target resource.

SP-Initiated SSO: Artifact/Artifact

In this scenario, the SP sends a SAML [artifact](#) to the IdP via an HTTP redirect. The IdP uses the artifact to obtain an authentication request from the SP. Then the IdP sends another artifact to the SP, which the SP uses to obtain the SAML response.

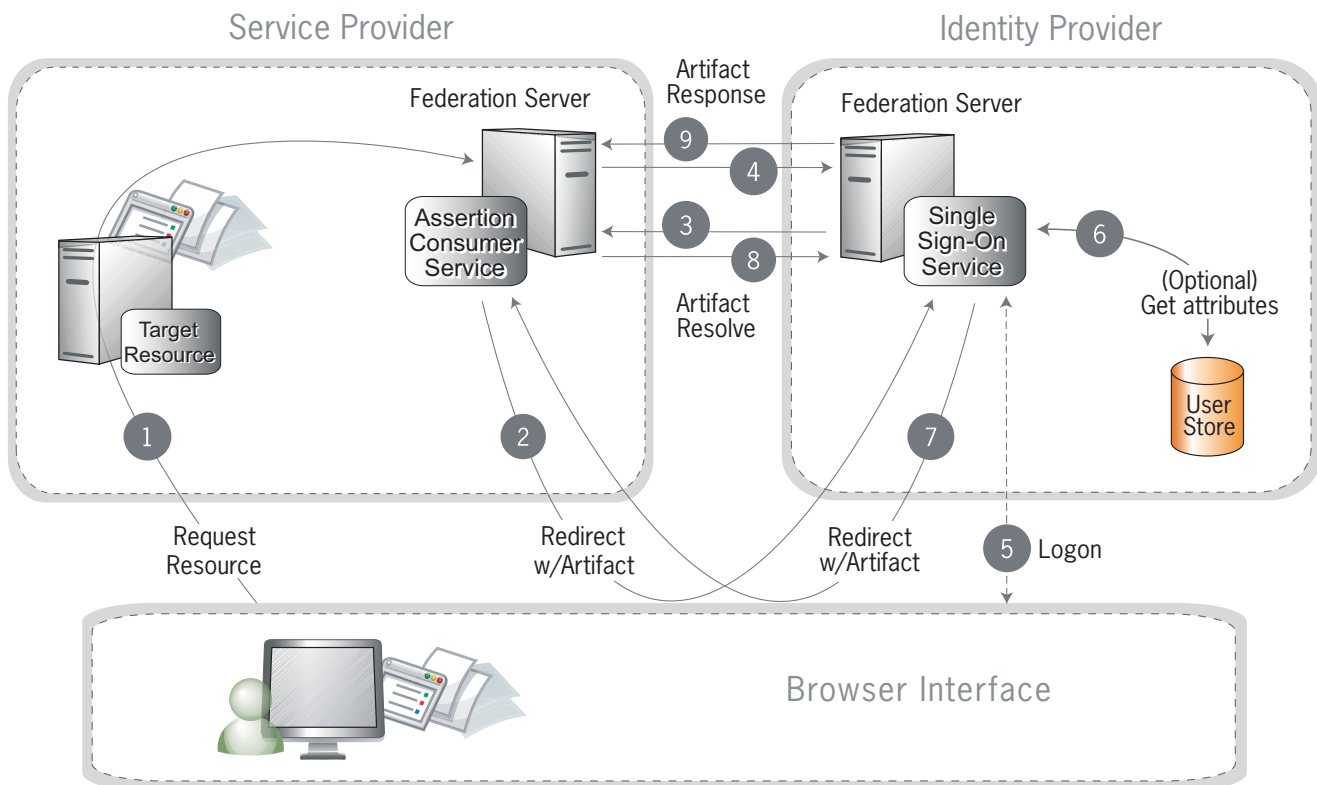


Figure 12: SP-Initiated SSO: Artifact/Artifact

Processing Steps:

1. A user requests access to a protected SP resource. The user is not logged on to the site. The request is redirected to the federation server to handle authentication.
2. The ACS generates an authentication request and creates an artifact. It sends an HTTP redirect containing the artifact through the user's browser to the IdP's SSO service.



Note: The artifact contains the source ID of the SP's artifact resolution service and a reference to the authentication request.

3. The SSO service extracts the source ID from the SAML artifact and sends a SAML artifact resolve message containing the artifact to the SP's artifact resolution service.



Note: The SP and IdP's source IDs and remote artifact resolution services are mapped according to the federation agreement prior to this action.

4. The SP's artifact resolution service sends back a SAML artifact response message containing the previously generated authorization request.
5. If the user is not already logged on to the IdP site or if re-authentication is required, the IdP asks for credentials (e.g., ID and password) and the user logs on.
6. Additional information about the user may be retrieved from the user data store for inclusion in the SAML response. (These attributes are predetermined as part of the federation agreement between the IdP and the SP—see [“About Attributes”](#) on page 40.)
7. The IdP federation server generates an assertion, creates an artifact, and sends an HTTP redirect containing the artifact through the browser to the SP's Assertion Consumer Service (ACS).
8. The ACS extracts the Source ID from the SAML artifact and sends an artifact-resolve message to the identity federation server's Artifact Resolution Service (ARS).
9. The ARS sends a SAML artifact response message containing the previously generated assertion.
10. (Not shown) If a valid assertion is received, the SP establishes a session and redirects the browser to the target resource.

IdP-Initiated SSO: POST

In this scenario, a user is logged on to the IdP and attempts to access a resource on a remote SP server. The SAML assertion is transported to the SP via HTTP POST.

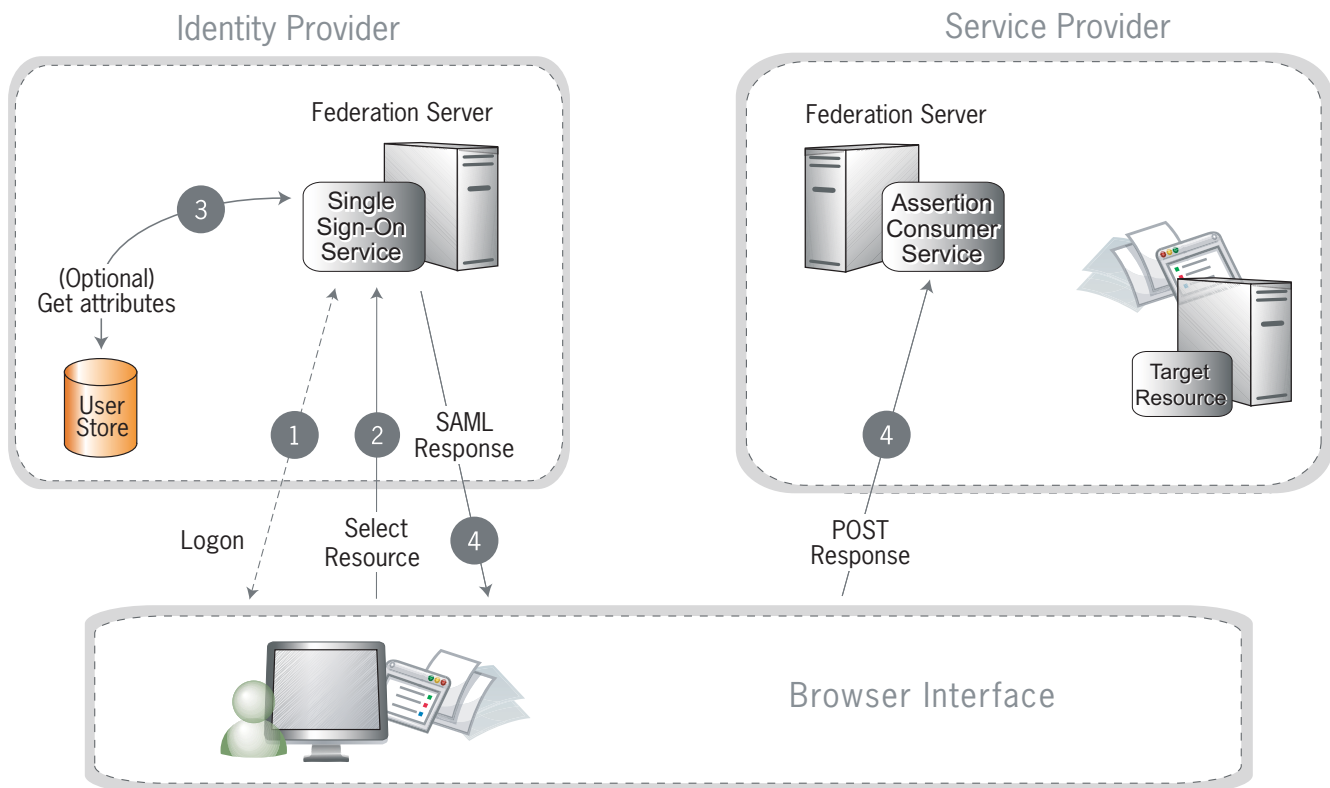


Figure 13: IdP-Initiated SSO: POST

Processing Steps:

1. A user has logged on to the IdP.
2. The user requests access to a protected SP resource. The user is not logged on to the SP site.
3. Optionally, the IdP retrieves attributes from the user data store.
4. The IdP's SSO service returns an HTML form to the browser with a SAML response containing the authentication assertion and any additional attributes. The browser automatically posts the HTML form back to the SP.



Note: SAML specifications require that POST responses be digitally signed.

5. (Not shown) If the signature and assertion are valid, the SP establishes a session for the user and redirects the browser to the target resource.

IdP-Initiated SSO: Artifact

In this scenario, the IdP sends a SAML artifact to the SP via an HTTP redirect. The SP uses the artifact to obtain the associated SAML response from the IdP.

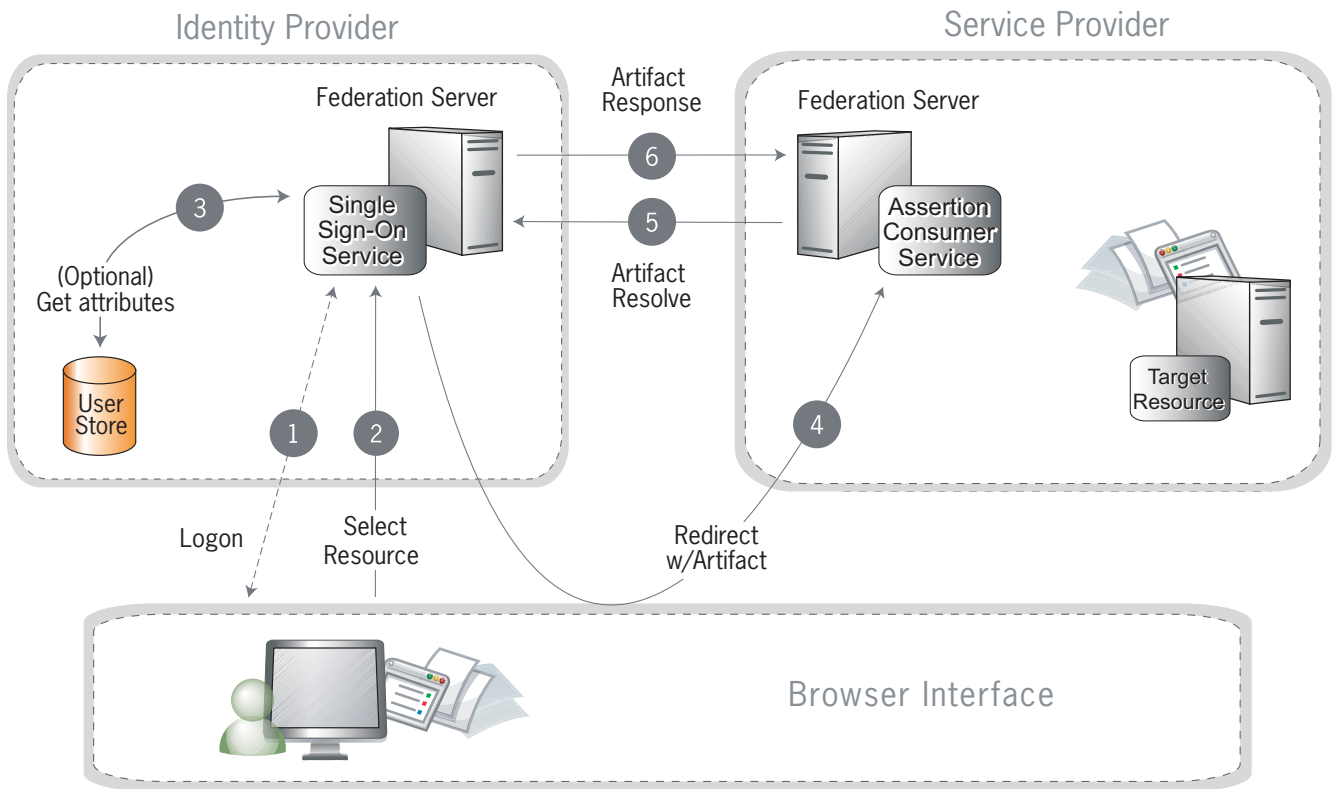


Figure 14: IdP-Initiated SSO: Artifact

Processing Steps:

1. A user is logged on to the IdP.
2. The user requests access to a protected SP resource. The user is not logged on to the SP site.
3. Optionally, the IdP retrieves attributes from the user data store.
4. The IdP federation server generates an assertion, creates an artifact, and sends an HTTP redirect containing the artifact through the browser to the SP's Assertion Consumer Service (ACS).
5. The ACS extracts the Source ID from the SAML artifact and sends an artifact-resolve message to the identity federation server's Artifact Resolution Service (ARS).
6. The ARS sends a SAML artifact response message containing the previously generated assertion.
7. (Not shown) If a valid assertion is received, the SP establishes a session and redirects the browser to the target resource.

Single Logout

The single logout (SLO) profile enables a user to log out of all participating sites in a federated session nearly simultaneously. The user may log out globally from any site, whether SP or IdP, as determined by respective Web applications. The associated IdP federation deployment handles all logout requests and responses for participating sites.

The logout messages may be transported using any combination of [bindings](#) described for SSO (POST, artifact, or redirect). Refer to the diagrams under “[Single Sign-on](#)” on page 20 for illustrations of these message flows.

About Session Clean-up

When an SP receives an SLO request from an IdP, the session creation [adapter\(s\)](#) you are using must handle any session clean-up with respect to the local application. For more information about adapters, see “[Integration Kits and Adapters](#)” on page 39.

Attribute Query and XASP

The SAML 2.0 Attribute Query profile allows an SP to request user attributes from an IdP in a secure transaction separate from SSO. The IdP, acting as an *Attribute Authority*, accepts Attribute Queries, performs a data-store lookup into a user repository such as an LDAP directory, provides values to the requested attributes, and generates an Attribute Response back to the originating SP requester. The SP then returns the attributes to the requesting application.



Tip: When privacy is required for sensitive attributes, you can configure PingFederate to obfuscate (mask) their values in the server and transaction logs (see “[Attribute Masking](#)” on page 42).

The X.509 Attribute Sharing Profile (XASP) is an OASIS draft standard that defines a specialized extension of the general Attribute Query profile. The XASP specification enables organizations with an investment in PKI (Public Key Infrastructure) identity management to issue and receive Attribute Queries based on user-certificate authentication.

Under XASP a user authenticates directly with an SP application by providing his or her X.509 certificate (see “[Application Authentication](#)” on page 120). Once the user is authenticated, the SP application requests additional user attributes from the SP PingFederate server. A portion of the user’s X.509 certificate is included in the request and used to determine the correct IdP to use as the source of the requested attributes (see “[Attribute Requester Mapping](#)” on page 208). Finally, the SP generates an Attribute Query and transmits it to the IdP over the SOAP back channel.

Because the user arrives at the SP server already authenticated, note that no PingFederate adapter is used in this use-case. Since the Web SSO use-case is distinct from the Attribute Query use-case, PingFederate servers may choose to implement one or both of these profiles without regard to the other.

At the time of this writing, the XASP is still in draft form in an OASIS committee. PingFederate implements the latest version of this draft and will be enhanced to continue adherence to the specifications, as needed. In the interim, configuration settings provided to a PingFederate administrator are flexible enough that minor changes in the specifications should not affect the server's compliance.

IdP Discovery

The SAML 2.0 IdP Discovery profile provides a mechanism by which SP Web application developers can look up a user's IdP rather than hard code that information. This mechanism can be helpful, in particular, in cases where an SP might be a hub for several IdPs in an identity federation.

In this scenario, when a user requests access to a protected resource on the SP, common-domain browser cookies are used to determine where a user has authenticated in the past. Using this information, a SAML 2.0 deployment such as PingFederate can determine which IdP connection to use for sending an authentication request.

As an IdP Discovery provider, PingFederate can serve in up to three different roles:

- Common domain server
- Common domain cookie writer
- Common domain cookie reader

Each of these roles is necessary to support IdP Discovery. The roles may be distributed across multiple servers at different sites.

Common domain server In this role the PingFederate server hosts a domain that its federation partners share in common. The common domain server allows partners to manipulate browser cookies that exist within that common domain. PingFederate can serve in this role exclusively or as part of either an IdP or an SP federation role, or both.

Common domain cookie writer When PingFederate is acting in an IdP and authenticates a user, it can write an entry in the common domain cookie, including its federation entity ID. An SP can look up this information on the common domain (not the same location as the common domain server described above).

Common domain cookie reader When PingFederate is acting as an SP and needs to determine the IdPs with whom the user has authenticated in the past, it reads the common domain cookie. Based on the information contained in the cookie, PingFederate can then initiate an SSO authentication request using the correct IdP connection.

WS-Federation

PingFederate supports the WS-Federation Passive Requestor Profile, enabling interoperability with Microsoft's Active Directory Federation Service (ADFS).

This profile provides for straightforward redirects and HTTP GET and POST methods to transport SAML assertions as security tokens for SSO and logout request and response messages for SLO.



Note: Unlike SAML, WS-Federation consolidates the endpoints for SLO and SSO. So when you set up a WS-Federation connection in PingFederate, both types of transactions are available to an SP Web application that supports them both.

For more information about WS-Federation and the Passive Requestor Profile, see “Web Services Federation Languages” at:

<http://www-128.ibm.com/developerworks/library/specification/ws-fed/>

Passive Requestor Profile

This profile permits a user’s browser (the passive requestor) to request a security token from an IdP when the user requests access to a protected Web service or other resource.

Figure 15 on page 33 illustrates message processing for SSO using WS-Federation.

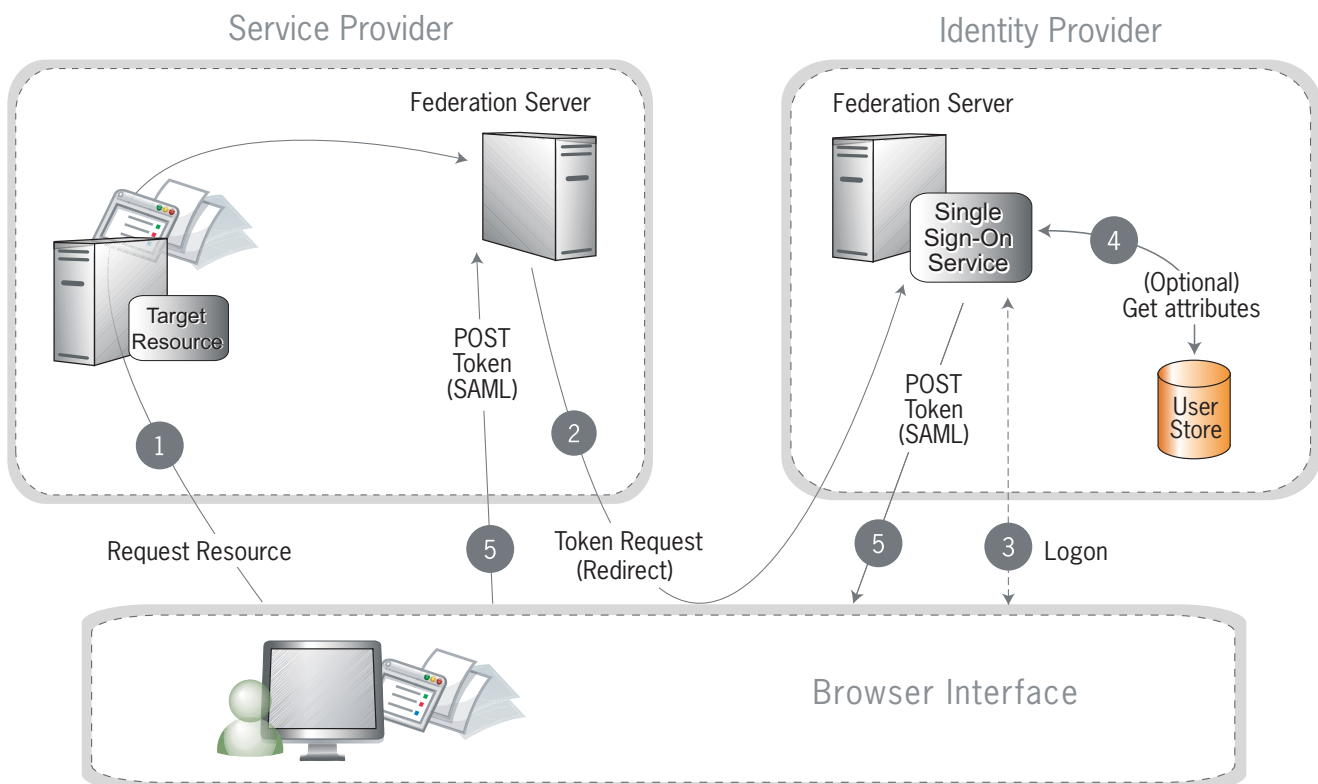


Figure 15: WS-Federation SSO

Processing Steps:

1. A user requests access to a protected SP resource. The user is not logged on to the site. The request is redirected to the federation server to handle authentication.
2. The SP generates a security token request and redirects the browser to the identity provider's WS-Federation implementation.
3. If the user is not already logged on to the IdP site or if re-authentication is required, the IdP asks for credentials (e.g., ID and password) and the user logs on.
4. Additional information about the user may be retrieved from the user data store for inclusion in the SAML response. (These attributes are predetermined as part of the federation agreement between the IdP and the SP—see [“About Attributes”](#) on page 40.)
5. The federation server creates a response containing a signed SAML assertion and returns it to the SP via POST.
6. (Not shown) If the signature and assertion are valid, the SP establishes a session for the user and redirects the browser to the target resource.

Single logout under WS-Federation is handled in much the same way as under SAML (see [“Single Logout”](#) on page 31); however, HTTP GET/POST is always used as the transport mechanism.

Account Linking

Account linking provides a means for a user to log on to disparate sites with just one authentication, when the user has established accounts and credentials at each site. This method of effectively interconnecting accounts across domains is supported by all protocols.

Account linking involves a *persistent name identifier* associated with accounts at each participating site. The name identifier, which may be an opaque [pseudonym](#), is conveyed in the [assertion](#). Once established locally, the SP can use the account link to look up the user and provide access without reauthentication.

For more information about account linking, see [“Account Linking”](#) on page 38.

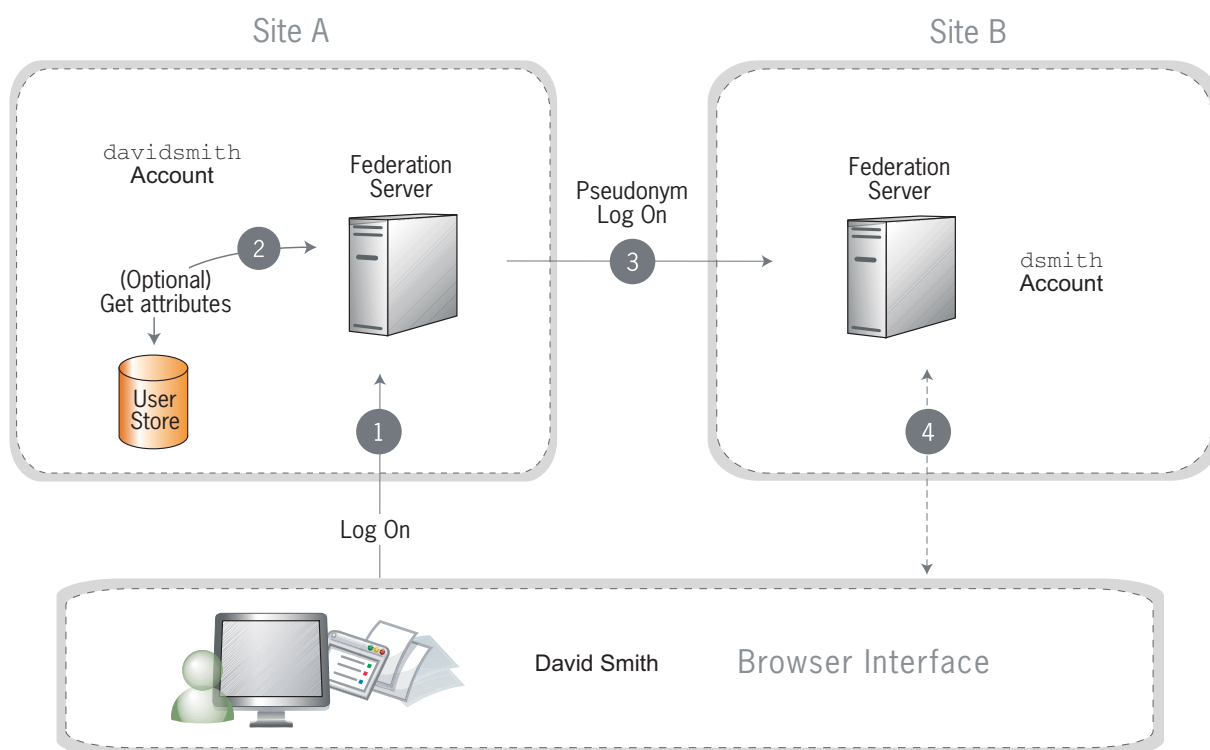


Figure 16: Account Linking

Processing Steps:

- David Smith logs on to Site A as **daVIDsmith**. He then decides to access his account on Site B via Site A.
- Optionally, the federation server looks up additional attributes from the data store.
- The Site A federation server sends a persistent name identifier (possibly a **pseudonym**) to Site B, along with any other attributes.
If a pseudonym is used and other attributes are sent, care must be taken not to send attributes that could be used to identify the subject.
- The federation server on Site B uses the information to associate the pseudonym with the existing account of **dsmith**. (Optionally, David is asked to provide consent to the linking.)

Once the link has been established, it is stored so that David only has to log on to Site A to have access to Site B.

Transport and Message Security

SAML defines two main ways of securing its interactions: Secure Sockets Layer with Transport Level Security (SSL/TLS) and digital signatures. SSL/TLS is used in environments where both message confidentiality and integrity are required.

Digital signatures are used to ensure the identity of both parties involved in the transaction and to validate that a message was received from a particular partner.

For more information, refer to [Security and Privacy Considerations for the OASIS Security Assertion Markup Language \(SAML\) V2.0](#) available on the [SSTC Web site](#).

Key Concepts

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

- [“Identity Mapping”](#) on page 37
- [“Integration Kits and Adapters”](#) on page 39
- [“About Attributes”](#) on page 40
- [“Certificates, SSL, and XML Encryption”](#) on page 43
- [“Federation Planning Checklist”](#) on page 46

Identity Mapping

Identity mapping is the conceptual core of identity federation. A user is generally known by different identifiers and roles in different security domains. One of the primary goals of the SAML [assertion](#) is to allow an IdP to provide a security token containing enough user-identity information that an SP can translate, or map, that information to local user stores.

PingFederate enables two modes of identity mapping between domains:

- [Account Linking](#)
- [Account Mapping](#)

Account Linking

Under SAML and WS-Federation, *account linking* can be used 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 294). 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 193). 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 will reduce 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 “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).

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.

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 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:

- A Standard Adapter, which provides a generic interface for integrating with various applications, including Java- and .NET-based applications (see [“Standard Adapter Configuration”](#) on page 267)
- An LDAP Authentication Service, which interfaces LDAP v3-supported active directories (see [“LDAP Adapter Configuration”](#) on page 279)

For a demonstration of the use of the Standard Adapter, see the PingFederate *Quick Start Guide* in the `/quickstart/docs` directory.

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 `Readme.txt` file in the `/sdk` directory for more information.

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. 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](#) to populate an attribute contract directly or use these attributes to look up additional attributes in IdP data stores (see “[Creating an Attribute Contract](#)” on page 149).
- Define and enforce a set of required attributes needed by SP adapters to interface local systems or applications (see “[Adapter Contracts](#)” on page 41).
- Set up connections to local data stores (see “[Data Stores](#)” on page 42).
- Configure specific attribute sources and lookups—based on the data stores—and map attributes into IdP assertions or into SP adapters used to interface target applications (see “[Integration Kits and Adapters](#)” on page 39).
- Selectively mask attribute values recorded in transaction logs (see “[Attribute Masking](#)” on page 42).

Attribute Contracts

An attribute contract represents an agreement between an SP and an IdP about user attributes sent in an [assertion](#). 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 42).

For an IdP, the attribute contract defines which attributes PingFederate sends in an assertion (see [“Single Sign-on”](#) on page 20). 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 three different sources of data:

- An IdP attribute source, which identifies the location of individual attributes in a data store (see [“Configuring Attribute Sources and User Lookup”](#) on page 154)
- The SP (see [“Integration Kits and Adapters”](#) on page 39)
- Static text values for some attributes or text values combined with variables (see [“Attribute Contract Fulfillment”](#) on page 165)

For an SP, the attribute contract defines the attributes PingFederate expects in an SSO assertion. PingFederate can be configured to pass these attributes to the SP adapter (see [“Integration Kits and Adapters”](#) on page 39). You can also use attributes to look up additional attributes in local data stores, which may be needed to start a user session (see [“Adapter Contracts”](#) below).

The attribute contract must contain the attribute `SAML_SUBJECT`, the primary information used to identify the user, unless you are using account linking. 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.

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 “last-mile” 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 [“Integration Kits and Adapters”](#) on page 39 and [“Configuring IdP Adapters”](#) on page 124).

On the SP side, adapter contract attributes are those required by adapters in order to start a session with applications. At least one *adapter type* is needed for each security domain. Then an *adapter instance* must be configured for each target application.

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 a Siteminder adapter type and configure an adapter instance for each protected application (see [“Configuring Adapter Mapping and User Lookup”](#) on page 228).

Extended Adapter Contract

Adapter contracts are created when the developer deploys an adapter type with PingFederate. At the time of deployment, 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*.

Data Stores

PingFederate can be configured to use local data stores to supply attributes for either the IdP’s [attribute contract](#) or SP’s [adapter contract](#) (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 77).

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 `README.txt` 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 90). 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 77).
- Retrieval of attributes from an IdP adapter (see [“Setting Pseudonym Values and Masking”](#) on page 128).

- SP-server processing of incoming attributes based on the SSO Attribute Contract, (see [“Creating an Attribute Contract”](#) on page 226).

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 38).

- SP-server processing of incoming attributes in response to an Attribute Request under XASP (see [“Specifying Security Policy”](#) on page 255).

For information about XASP, see [“Attribute Query and XASP”](#) on page 31.



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 `log4j.xml` to prevent the recording of attributes (see [“Log File Generation”](#) on page 90).

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 111).

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 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, OASIS recommends digital signing practices, using public/private keypairs in conjunction with X.509 certificates.



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.)

PingFederate enables signing and validation of responses, requests, and/or the assertion message. In addition, PingFederate comes with embedded 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 118).



Note: Ping Identity recommends generating unique certificates for each connection, which limits your exposure if your private key were to become 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. 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

When you import a certificate into PingFederate, the system checks to make sure that the certificate has not expired. This check is also made at runtime, when a certificate is used.

PingFederate may also check Certificate Revocation Lists (CRLs) to ensure that a certificate has not been revoked. At runtime, the server can perform this check on signature verification certificates. The server can check CRLs and honor a revocation only if the follow conditions are met:

- The certificate is signed by a CA contained in the PingFederate trusted store (see [“Trusted CAs”](#) on page 111).
- The certificate contains the URL where the CA maintains its CRL.
- The CRL's URL is accessible to PingFederate.
- The CRL is signed and the signature verified.

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, expired or revoked certificates are listed in red.

Digital Signing Policy Coordination

To coordinate digital signature policy, partners must first agree about whether they will sign SAML messages. In some cases, the protocol specifications require

signatures—for example, all 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. Numerous scenarios are possible, including:

- SP verifies incoming response signatures
- SP verifies incoming assertion signatures
- SP signs outgoing requests
- IdP signs outgoing responses
- IdP signs outgoing assertions (for any binding)
- IdP sign outgoing requests for single logout
- IdP verifies incoming request signatures

The signing partner must send certificates (containing only the public keys) out-of-band to the validating partner, who must import the certificates into PingFederate before they can be used for validation of signed messages (see [“Digital Signing and Decryption Keys & Certificates”](#) on page 118).

Example Configuration Scenario: IdP Signs Outgoing Response

1. The IdP generates a private and public keypair and submits a certificate for authority signing (optional: self-signing by the IdP).
2. The IdP imports the CA’s signing response certificate into the PingFederate keystore.
3. The IdP configures the connection to the SP to sign SAML responses.
4. IdP exports the public key of the certificate, which is sent to the SP out-of-band.
5. The SP administrator imports the certificate into the PingFederate digital signature verification keystore for the IdP federated connection.
6. The SP administrator configures the connection to use the certificate to verify the digital signature on incoming IdP responses?

Secure Sockets Layer

SSL certificates signed by a certificate authority 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 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.

Authentication

PingFederate can be configured to use SSL/TLS to encrypt the communication channel over which SAML transactions are transmitted and to authenticate connection partners making SOAP requests. Three methods of authentication are available for use with PingFederate. 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 115).
- Digital Signatures: partners sign the XML message transmitted via the SSL/TLS connection. Signatures are verified by the receiver based upon the certificates 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 118).

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 Certificate Authority (CA), by which a certificate is issued, must be imported into PingFederate's trusted certificate store.

XML Encryption

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

For SAML 2.0 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.

Federation Planning Checklist

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

Legal Agreements

Ensure legal agreements are in place that reflect specific configuration guidelines agreed upon, limits of liability, and so on. Contact [Ping Identity](#) for a useful handbook about these issues.

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 43.)

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 certificate authority. (If self-signed certificates or nonstandard certificate authorities are to be used, the signed certificates must be exchanged and imported into Trusted Certificate stores.)

Deployment

Decide how PingFederate fits into your existing network. A deployment within a DMZ may suit your requirements (see [“Deployment Options”](#) on page 56). Also, determine whether high-availability and/or failover options are required (see [“Clustering and Failover Deployment”](#) on page 295).

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 49).

With PingFederate, you define a unique ID for each supported protocol (see [“Specifying Federation Information”](#) on page 75). 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 139) or as an SP configuring an IdP connection (see [“General Information”](#) on page 217).



Tip: PingFederate also provides for *virtual host names*, which are different than virtual IDs (but 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 105).

Server Clock Synchronization

Ensure that both the SP and IdP server clocks are synchronized. SAML messages 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 42).

Web Application and Session Integration

Decide how the IdP side of PingFederate receives subject identity information to look up the session.

For an SP, decide how PingFederate will forward user identity information to the destination Web application or system to start a session. (See [“Integration Kits and Adapters”](#) on page 39).

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 90.)

Identity Mapping

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 37).

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 more information, see [“Attribute Contracts”](#) on page 41.)

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 federation.

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 47).

- **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).

- **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 47).

- **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 40).
- **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.

Installation

PingFederate is packaged as a stand-alone server based on J2EE application server technology.

This chapter covers:

- [“System Requirements”](#) on page 52
- [“Installing the JDK”](#) on page 53
- [“Installing PingFederate”](#) on page 54
- [“Running PingFederate for the First Time”](#) on page 54
- [“Deployment Options”](#) on page 56
- [“Installing PingFederate as a Service”](#) on page 58
- [“Uninstalling PingFederate”](#) on page 60

System Requirements

PingFederate is supported for deployment and configuration with the system specifications below.



Note: PingFederate should function normally under a variety of platform configurations, including Web browsers, not specified below. Also, data stores may potentially include any LDAP v3-compatible directory service or JDBC-compatible database.

Platform and data-store testing and qualification are ongoing; consult the PingFederate product [Web site](http://www.pingidentity.com/products/pingfederate/download) (www.pingidentity.com/products/pingfederate/download) for the latest information.

Operating Systems

Windows

- Microsoft Windows Server 2003 with Service Pack 1 on x86 (32-bit)
- Windows XP Professional with Service Pack 2 (32-bit)

Linux

- Red Hat Enterprise Linux ES 4 with 2.6.9-42.0 Kernel on x86 (32-bit)



Note: PingFederate has been tested with default configurations of operating system components. If your organization has customized implementations or has installed third-party plug-ins, deployment of the PingFederate server may be affected.

User Store/Data Store Integration

JDBC Compatible

- Oracle9i - 9.2.0.1 (on Windows 2003)

LDAP v3 Compatible

- Active Directory 2003 (with SP 1)
- Sun One Directory Server 5.2

Browsers

- Internet Explorer 7.0
- Firefox 2.0

Browsers must be JavaScript-enabled.

Java Environment

JDK 1.5 or higher



Important: The JDK must be installed in a path containing no spaces (for example, do *not* use the “Program Files” folder on Windows).

Minimum Hardware Requirements

- Intel Pentium 4, 1.8 GHz processor
- 512 MB of RAM
- 250 MB of available hard drive space

Installing the JDK

The J2SE Development Kit (JDK) provides the required environment for PingFederate.



Important: You must install JDK before installing PingFederate.

To Install the J2SE Development Kit for Windows and Linux:

1. Download the JDK at: <http://java.sun.com/j2se/1.5.0/download.jsp>.
2. Install the JDK to a location with no spaces in the path (for example, C:\j2sdk1.5).
3. Set the JAVA_HOME environment variable to the JDK installation directory path and add the /bin directory to the PATH variable for your platform.



Note: If running PingFederate as a service, you must set JAVA_HOME at the system level.

Installing PingFederate

You install PingFederate by extracting the `pingfederate-4.3.x.zip` file.



Note: If your site requires compliance with FIPS 104-2, see [“Using the SafeNet Luna HSM”](#) on page 307 for additional installation information.



Important: On Unix or Linux you must install and run PingFederate under a local user account.

To install PingFederate:

1. Ensure you are logged into your system with appropriate privileges to install and run an application.
2. Verify that the JDK is installed and environment and PATH variables are set correctly (see [“Installing the JDK”](#) on page 53).
3. Create an installation directory.



Important: The installation path and the directory name must *not* contain spaces.

4. Unzip `pingfederate-4.3.x.zip` into the installation directory.
5. Request a license key.

Sign on to the Ping Identity Web licensing page (<http://www.pingidentity.com/support/licensing>).

6. Save the license key file in the directory:

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

Ensure the file is named:

`pingfederate.lic`



Note: If you are deploying PingFederate in a cluster configuration, you may install the license key on any server in the cluster. (For more information, see [“Clustering and Failover Deployment”](#) on page 295.)

Running PingFederate for the First Time

The first time you run PingFederate, you use a default username and password supplied with the distribution. After you launch the administrative console and log on, you must change the password. After that, “post-installation” screens guide you through an initial setup process, during which you may choose the

federation protocols you will use and enter information necessary to configure your federation role (IdP, SP, or both).



Tip: As part of the post-installation process, you are asked to configure at least one [adapter](#) for PingFederate to use with your Web application(s) or identity management system (see [“Integration Kits and Adapters”](#) on page 39). If you do not yet know parameters for this configuration, enter placeholders during post-installation (see the online **Help** for the configuration screens). Or refer to the *Quick Start Guide* in the `quickstart/docs` directory for instructions in setting up a sample configuration.

When the post-installation process is complete, the Main Menu opens.



Note: You can change the post-installation setup via menu choices under My Server on the Main Menu (see [“System Settings”](#) on page 69).

To run PingFederate for the first time:

1. Start the PingFederate server by running the following script:

(Windows) `<PF_install_dir>/pingfederate/bin/run.bat`

(Linux) `<PF_install_dir>/pingfederate/bin/run.sh`

Wait for the script to finish the startup—the last message displayed in the sequence is:

Started in XXs:XXms

2. Launch your browser and go to `https://<DNS_NAME>:9999/pingfederate/app`.

Where `<DNS_NAME>` is the fully qualified name of the machine running the PingFederate server.



Note: The port number 9999 is set by default. For information on changing this setting, see [“Changing Configuration Parameters”](#) on page 107.

3. Enter the default Username and Password:

Username: `Administrator`

Password: `2Federate`

4. Change your password on the Change Password screen and click **Save**.



Note: The new password must be at least six characters and contain at least one uppercase character, one lowercase character, and one numeric character.



Important: Take steps to ensure that you do not forget the new password. For more information about passwords and user management, see [“Account Management”](#) on page 101.

5. Complete the steps in the Configuring My Server screens.

For more information see sections under [“Managing Server Settings”](#) on page 69.

Deployment Options

There are many options for deploying PingFederate in your network environment, depending on your needs and infrastructure capabilities.

For example, you can choose a stand-alone or proxy configuration, as described in this section. Or you can deploy multiple PingFederate servers in a cluster configuration for high availability, server redundancy, and failover recovery (see [“Clustering and Failover Deployment”](#) on page 295).

[Figure 17](#) illustrates PingFederate installed in the DMZ.

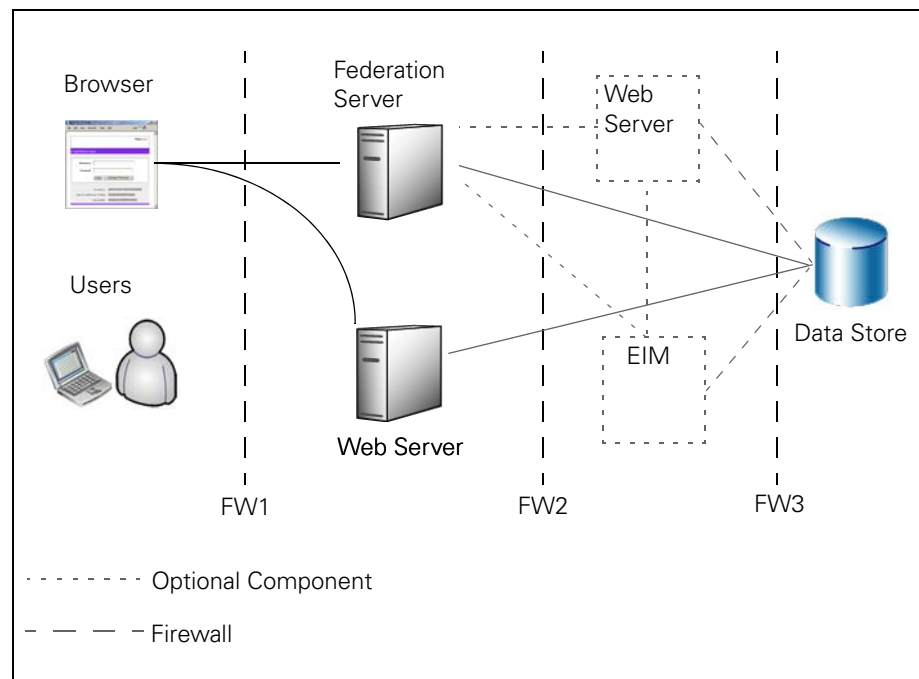


Figure 17: Stand-alone Deployment Example

In this configuration, users access PingFederate via a Web application server (and/or an Enterprise Identity Management system). PingFederate may, in turn, retrieve information from a data store to use in processing the transaction.

You can also deploy PingFederate with a proxy server. [Figure 18](#) depicts a proxy-server configuration in which the proxy is accessed by users and Web browsers. The proxy, in turn, communicates with PingFederate to request SSO.

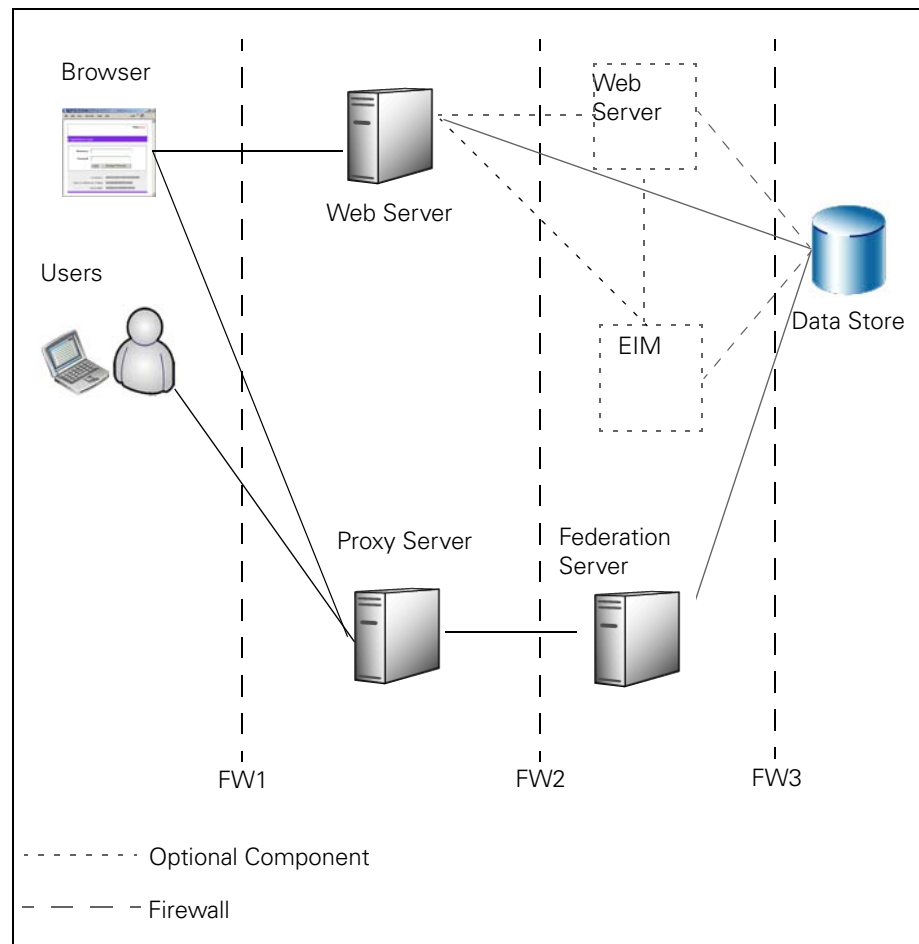


Figure 18: Proxy Deployment Example

Installing PingFederate as a Service

You can set up PingFederate to run in the background as a service on either Windows or Linux.



Note: Before performing this procedure, ensure that PingFederate runs normally by manually starting the server (see [“Running PingFederate for the First Time”](#) on page 54).

(Windows)

This installation enables PingFederate to start automatically when Windows is started or rebooted.

To install PingFederate as a Service:

1. Complete the steps under [“Installing PingFederate”](#) on page 54.
2. From a command prompt in the `<PF_install_dir>/pingfederate/sbin` directory, execute the `install-pf-svc.bat` file.
3. A message confirms the installation.
4. Access the Windows **Control Panel > Administrative Tools > Services**.
5. Right-click PingFederate from the list of available services and select **Properties**.
6. Select the **General** tab, and click **Start**.
7. (Optional) Set the service to automatically start at Windows start up.

(Linux)

To install PingFederate as a service on Linux, you must place a script in the system initialization directory. Before running PingFederate as a service, manually start the PingFederate server to ensure that it is configured properly (see [“Running PingFederate for the First Time”](#) on page 54).



Note: If you are not using RedHat, you may need to modify references to the system initialization directory in this procedure—for example, Debian uses `/etc/init.d/` instead of `/etc/rc.d/init.d/`.

To run PingFederate as a Linux service (RedHat):

1. Complete the steps under [“Installing PingFederate”](#) on page 54.
2. Log on as `root`.
3. Create a new user account for the service.

For this procedure, the variable `<PF_user>` is used to refer to this account.

4. Change the PingFederate installation directory (<PF_install_dir>) ownership and ensure its read/write property:

```
chown -R <PF_user> <PF_install_dir>
chmod -R 775 <PF_install_dir>
```

5. Place the code below into a file called <PF_user> in the directory: /etc/rc.d/init.d/



Note: Replace instances of <PF_user> and <PF_install_dir> in the script below, and in the commands that follow, with their respective values.

This script, modified to work with PingFederate, is based on *StartJBossOnBootWithLinux* script at the <http://www.jboss.com/> Web site.

```
#!/bin/sh

start(){
    echo "starting PingFederate.."
    su - <PF_user> -c '<PF_install_dir>/sbin/pingfederate-
run.sh > /dev/null 2> /dev/null'
}

stop(){
    echo "stopping PingFederate.."
    su - <PF_user> -c '<PF_install_dir>/sbin/pingfederate-
shutdown.sh -S'
}

restart(){
    stop
    # padding time to stop before restart
    sleep 60
    # protect against any services that are not stopped
    # (warning: this kills all Java instances running as
    # <PF_user>')
    su - <PF_user> -c 'killall java'
    start
}

case "$1" in
    start)
        start
        ;;
    stop)
        stop
        ;;
    restart)
        restart
        ;;
    *)
```

```
        echo "Usage: <PF_user> {start|stop|restart}"
        exit 1
    esac
    exit 0
```

6. Create symbolic links using commands listed below.

The links specify the order in which the PingFederate Server starts and stops.

```
ln -s /etc/rc.d/init.d/<PF_user> /etc/rc3.d/S84<PF_user>
ln -s /etc/rc.d/init.d/<PF_user> /etc/rc5.d/S84<PF_user>
ln -s /etc/rc.d/init.d/<PF_user> /etc/rc4.d/S84<PF_user>
ln -s /etc/rc.d/init.d/<PF_user> /etc/rc6.d/K15<PF_user>
ln -s /etc/rc.d/init.d/<PF_user> /etc/rc0.d/K15<PF_user>
ln -s /etc/rc.d/init.d/<PF_user> /etc/rc1.d/K15<PF_user>
ln -s /etc/rc.d/init.d/<PF_user> /etc/rc2.d/K15<PF_user>
```

7. Make the script executable (as root):

```
chmod 755 /etc/rc.d/init.d/<PF_user>
```

8. Test the script by entering:

```
service <PF_user> start
```

and then:

```
service <PF_user> stop
```

9. To start the service, enter:

```
service <PF_user> start
```

Uninstalling PingFederate

To uninstall PingFederate:

1. If PingFederate is installed as a service, follow the platform-specific procedure in the next section, [“Uninstalling Services”](#).
2. Delete the PingFederate installation directory.

Uninstalling Services

To uninstall PingFederate as a Windows Service:

1. Access the Windows **Control Panel > Administrative Tools** and double-click **Services**.
2. Right-click PingFederate from the list of available services and select **Properties**.
3. Select the **General** tab, and click **Stop**.
4. Execute `uninstall-pf-svc.bat` in the `\pingfederate\sbin` directory.

To uninstall PingFederate as a Linux Service:

1. Log on as root.

2. Stop the service with the command:

```
service <PF_user> stop
```

where <PF_user> is the PingFederate service user account (see [“Installing PingFederate as a Service”](#) on page 58).

3. Remove symbolic links:

```
rm /etc/rc3.d/S84<PF_user>
rm /etc/rc4.d/S84<PF_user>
rm /etc/rc5.d/S84<PF_user>
rm /etc/rc0.d/K15<PF_user>
rm /etc/rc1.d/K15<PF_user>
rm /etc/rc2.d/K15<PF_user>
rm /etc/rc6.d/K15<PF_user>
```

4. Optionally, delete the script used to start and stop the service (see [“Installing PingFederate as a Service”](#) on page 58).

Console Navigation

The PingFederate administrator's user interface, the *administrative console*, is built around a system of wizard-like control screens, which are accessed from a top-level portal, the Main Menu.

This chapter covers the use of these navigational features:

- [“Using the Main Menu”](#) on page 63
- [“Navigating the Administrative Console”](#) on page 65

Using the Main Menu

When you log on to PingFederate, you reach the Main Menu from which you can modify your local server settings or access configuration screens to set up or modify connections with partners (see [Figure 19](#) on page 64).



Note: This information is presented from the viewpoint of an administrative user with full permissions to configure local server settings and partner connections (see [“Account Management”](#) on page 101).

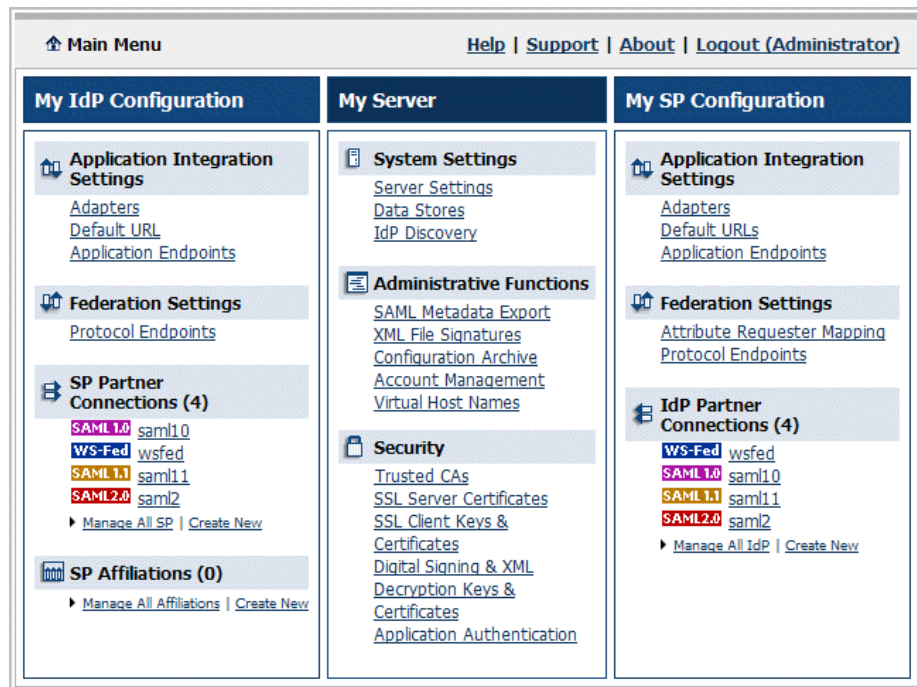


Figure 19: Main Menu (Example)

Note that Main Menu selections depend on your federation role (IdP, SP, or both) and which protocol(s) you are using (see “[Choosing Roles and Protocols](#)” on page 74). Selections also depend on your system permissions (see “[Account Management](#)” on page 101).

Depending on your permissions, you can use the Main Menu to:

- Modify or add to system settings after installation—see “[System Settings](#)” on page 69
- Handle system administration functions—see “[System Administration](#)” on page 89
- Manage certificates and application authentication for attribute query requests—see “[Security Management](#)” on page 111
- Configure connections and other IdP or SP settings—see “[Identity Provider Configuration](#)” on page 123 or “[Service Provider Configuration](#)” on page 197

Navigating the Administrative Console

PingFederate's configuration screens are designed to guide you through the process of setting up and maintaining your server. This configuration design provides three major benefits:

First, given the complicated security considerations and elaborate requirements under the SAML specifications, setting up an identity federation is complex. The PingFederate setup screens provide a step-by-step mechanism that minimizes the chance of overlooking critical settings.

Second, setting up a federation involves many choices based on your agreement with your partner (see [“Federation Planning Checklist”](#) on page 46). PingFederate presents these choices in an organized way and then takes you along the right path, presenting only the steps you need to take based on previous choices.

Finally, like most complex network configurations, federation setup involves many interdependencies. PingFederate keeps track of these for you: when you make a change, the system finds related changes and takes you to the relevant screens.



Caution: Do not use the browser's Back, Refresh, or Forward buttons. Instead, use the navigation buttons in the lower right portion of the configuration screens (see [“Console Buttons”](#) on page 67).

About Tasks and Steps

Each broad configuration area is broken down into a series of tasks. Each task consists of a sequence of steps. The tasks and steps appear in the top portion of the screen, as shown in [Figure 20](#).

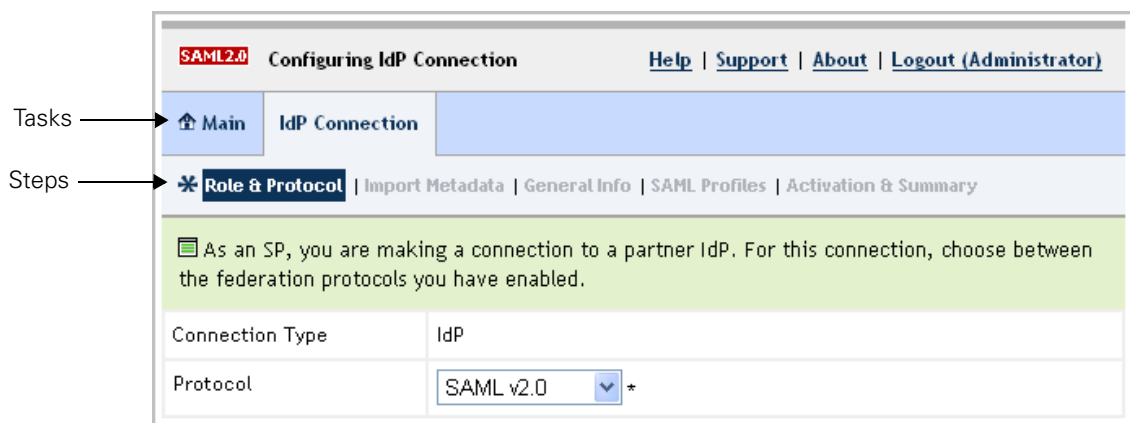


Figure 20: Tasks and Steps (Example)

Notice that steps you have not yet reached are grayed out. After you complete a step, you can click it to go back. When all the steps are completed, you can click any of them to review your work or make changes.



Important: Be sure to click **Save** when you reach the last step of a task (if you want to save changes), or if you have finished editing a step.

As you traverse steps for each task, you will notice that some steps branch to separate tasks used to accomplish a series of substeps. In addition, on some screens buttons provide shortcuts to other tasks, typically those used for global settings—for example, as you set up a connection to a partner, you might need to import a certificate into your trusted store (see “[Trusted CAs](#)” on page 111).

In either case, when you change tasks, the transitional step or related global task appears as the current task, and the steps change accordingly.



Caution: Clicking **Cancel** on any screen discards all new unsaved entries for all steps shown for the current task and returns you to the screen from which you accessed the task.

Console Buttons

The navigational and control buttons at the bottom of the administrative console screen change depending on where you are in the configuration process. The following table describes the behavior of these buttons.

Table 2: Administrative Console Buttons

Button	Description
Save	Stores information for all steps completed for the current task or any changes made for the current step; returns to the screen from the which the task or step was accessed. This button is available only when the Save operation is valid within the current context.
Done	Marks as complete all steps for a current task, but does not save the configuration (because further tasks or steps are necessary); to save entries or changes, continue the configuration until you see a Save button or click Save Draft (see below).
Save Draft	Stores a new connection configuration for all steps completed up to the current screen in the configuration flow. To return to the draft, click Manage All [IdP or SP] under [IdP or SP] Connections on the Main Menu and then select the draft from the connection list.
Cancel	Returns to the screen from which the current task was accessed; discards any information newly entered or modified for all steps in the task.
Previous	Returns to the previous step (when applicable).
Next	Moves display forward to the next step (when applicable), if all required information is complete in the current step.

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 69
- [“Managing Data Stores”](#) on page 77
- [“Configuring IdP Discovery”](#) on page 86



Note: The information in this chapter is presented from the viewpoint of an administrative user with “Admin” permissions (see [“Account Management”](#) on page 101).

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 [“Standards Support”](#) on page 13).

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).

You configure server settings initially during post-installation setup (see [“Running PingFederation for the First Time”](#) on page 54), but you can change them as needed.

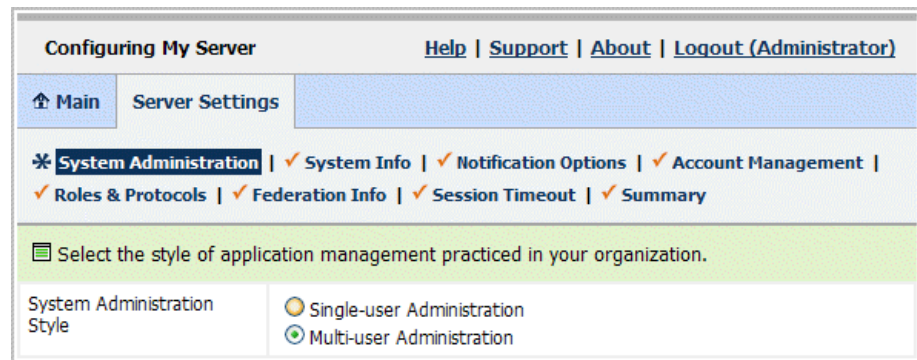
Information in this section covers:

- [“Setting Administration Style”](#) on page 70
- [“Entering System Information”](#) on page 71
- [“Configuring Notification Options”](#) on page 71
- [“Managing Accounts”](#) on page 73
- [“Choosing Roles and Protocols”](#) on page 74
- [“Specifying Federation Information”](#) on page 75
- [“Changing Session Timeout”](#) on page 76

Setting Administration Style

PingFederate provides a choice of single- or multi-user system administration. If you choose single-user administration, the administrative console is accessible only by using the default Administrator ID, for which full privileges are provided by default.

If you choose multi-user administration, the system provides role-based access control (see [“Account Management”](#) on page 101).



The screenshot shows the 'Configuring My Server' web interface. At the top, there are links for 'Help', 'Support', 'About', and 'Logout (Administrator)'. Below this is a navigation bar with 'Main' and 'Server Settings'. The 'Server Settings' section contains a list of steps: 'System Administration' (selected with a star icon), 'System Info', 'Notification Options', 'Account Management', 'Roles & Protocols', 'Federation Info', 'Session Timeout', and 'Summary'. Below the steps list, a green box contains the instruction: 'Select the style of application management practiced in your organization.' Underneath, there are two radio button options: 'Single-user Administration' (which is selected) and 'Multi-user Administration'.

To reach this screen:

1. Click **Server Settings** on the Main Menu.
2. Click **System Administration** in the steps list.

To set single- or multi-user administration:

- Make your selection and click **Next** or **Save** (if changing the setting).



Note: In order to select single-user administration, you must have only one user marked as active (see [“Account Management”](#) on page 101).

Entering System Information

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

The screenshot shows the 'Configuring My Server' interface. At the top, there are links for 'Help', 'Support', 'About', and 'Logout (Administrator)'. Below this is a navigation bar with 'Main' and 'Server Settings'. The 'System Info' tab is selected, indicated by a star icon. Below the navigation bar, there are several status indicators: 'System Administration' (checked), 'System Info' (starred), 'Notification Options' (checked), 'Account Management' (checked), 'Roles & Protocols' (checked), 'Federation Info' (checked), 'Session Timeout' (checked), and 'Summary' (checked). A green box contains the text: 'This is general information that identifies your server. This information is included whenever you export connection metadata.' Below this, there are four input fields: 'Company' (My Company), 'Contact Name' (My Name), 'Contact Number' (empty), and 'Contact Email' (my.name@mycompany.com).

To reach this screen:

1. Click **Server Settings** on the Main Menu.
2. Click **System Info** in the steps list.

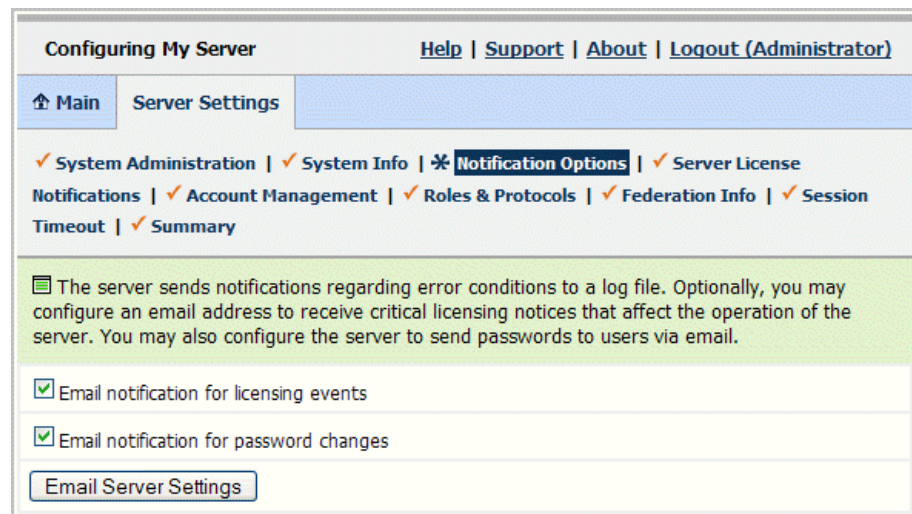
Configuring Notification Options

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

Also on the Notification Options screen, if you have chosen multi-user administration, you can choose to use email to notify users automatically of their new or reset passwords (see [“Setting Administration Style”](#) on page 70 and [“Account Management”](#) on page 101).



Note: If are setting up email notifications for the first time, then you must click **Email Server Settings** and configure the settings to continue (see [“Managing Email Configuration”](#) on page 100). Enter placeholders on the Email Notification screen if you are not sure of the correct settings during post-installation.



The screenshot shows a web interface titled "Configuring My Server". At the top right are links for "Help", "Support", "About", and "Logout (Administrator)". Below the title bar is a navigation menu with "Main" and "Server Settings". Under "Server Settings", there is a list of options: "System Administration", "System Info", "Notification Options" (which is highlighted with a blue background and a star icon), "Server License", "Notifications", "Account Management", "Roles & Protocols", "Federation Info", "Session Timeout", and "Summary". Below this list, a green box contains text: "The server sends notifications regarding error conditions to a log file. Optionally, you may configure an email address to receive critical licensing notices that affect the operation of the server. You may also configure the server to send passwords to users via email." Below this text are two checkboxes, both of which are checked: "Email notification for licensing events" and "Email notification for password changes". At the bottom of this section is a button labeled "Email Server Settings".

To reach this screen:

1. Click **Server Settings** on the Main Menu.
2. Click **Notification Options** in the steps list.



Note: If you have a perpetual license and you chose single-user administration at the System Administration step, then the Notification Options screen is not presented.

License Notification Address

If you choose to be notified via email about the status of your PingFederate license, the Server License Notifications screen is where you specify the target email address for notification.



Note: Notifications may include warnings when license expiration dates are approaching, as well as other important alerts, depending on your license agreement.

To reach this screen:

1. Click **Server Settings** on the Main Menu.
2. Click **Server License Notifications** in the steps list.

If this step does not appear, you have not enabled licensing notification (see “[Configuring Notification Options](#)” on page 71).



Note: The Server License Notifications screen is displayed only if your license is restricted. Perpetual licenses do not require notifications.

Managing Accounts

When you choose multi-user system administration, you can set up users during post-installation or while configuring Server Settings (see “[Setting Administration Style](#)” on page 70).

User Name	User Admin	Admin	Crypto Admin	Action
Administrator	<input type="radio"/> Auditor <input checked="" type="radio"/> Admin	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Deactivate / Change Password

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 101). By default for post-installation, the user “Administrator” has full system permissions.

- To continue, click **Next** or **Save**.

- For information about adding or managing users, see “[Account Management](#)” on page 101.

Choosing Roles and Protocols

At this step in the post-installation (or Server Settings) setup, you select which identity federation role(s) your organization plays and which set of standards you intend to use with your federation partner (see “[Federation Roles](#)” on page 13 and “[Standards Support](#)” on page 13).

Configuring My Server [Help](#) | [Support](#) | [About](#) | [Logout \(Administrator\)](#)

[Main](#) | **Server Settings**

✓ [System Administration](#) | ✓ [System Info](#) | ✓ [Notification Options](#) | ✓ [Account Management](#) |
✱ **[Roles & Protocols](#)** | ✓ [Federation Info](#) | ✓ [Session Timeout](#) | ✓ [Summary](#)

Select the role(s) and protocol(s) that you intend to use with your federation partners.

☒ Enable Identity Provider (IdP) role

- ☒ Enable SAML v2.0 protocol
- ☒ Enable SAML v1.1 protocol
- ☒ Enable SAML v1.0 protocol
- ☒ Enable WS-Federation protocol

☒ Enable Service Provider (SP) role

- ☒ Enable SAML v2.0 protocol
- ☒ Enable SAML v1.1 protocol
- ☒ Enable SAML v1.0 protocol
- ☒ Enable WS-Federation protocol

☒ Enable IdP Discovery role (SAML v2.0 only)

- Select your federation role(s) to see protocol selections, then make your protocol selection(s) and click **Next**.
You must choose at least one protocol for a selected role.
- Or if you are just 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:

1. Click **Server Settings** on the Main Menu.
2. Click **Roles and Protocols** in the steps list.

Specifying Federation Information

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



Note: You must provide an ID that uniquely identifies your federation gateway for each protocol you support. 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 47).

Configuring My Server
[Help](#) | [Support](#) | [About](#) | [Logout \(Administrator\)](#)

[Welcome](#) | [Additional Resources](#) | [Licensing](#) | [System Administration](#) | [System Info](#) | [Notification Options](#) | [Account Management](#) | [Roles & Protocols](#) | **[Federation Info](#)** | [Session Timeout](#) | [Summary](#)

You must create a unique identifier for your server for use with your federation partners. A unique identifier is required for each protocol enabled. You will need to communicate this with your partners out-of-band or through metadata exchange. The Base URL is used to construct other URLs in the system and may be used as part of your system ID.

Base URL	<input type="text" value="https://tyoes.corp.pingidentity.com"/> *
SAML v2.0 Entity ID	<input type="text" value="tyoes"/> *

Field Descriptions

Field	Definition
Base URL	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 94).

Field	Definition
SAML v2.0 Entity ID	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: <code>pingidentity.com</code> . The SAML SourceID used for artifact resolution is derived from this ID using SHA1.
SAML v1.x ID	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.
SAML v1.x Source ID	(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.
WS-Federation Realm	The URI of the realm associated with the PingFederate server. A realm represents a single unit of security administration or trust.



Note: The fields available on this screen depend on the federation protocols you have chosen to support (see [“Choosing Roles and Protocols”](#) on page 74).

To reach this screen:

1. Click **Server Settings** on the Main Menu.
2. Click **Federation Info** in the steps list.

Changing Session Timeout

If you are using the SAML 2.0 Single Logout Profile, PingFederate must keep track of user sessions to manage the logout process (see [“Single Logout”](#) on page 31). By default, if a user is inactive for one hour, the session will be discontinued.

You can change the default timeout on the Session Timeout screen under Server Settings.

Configuring My Server [Help](#) | [Support](#) | [About](#) | [Logout \(Administrator\)](#)

[Main](#) | **Server Settings**

✓ System Administration | ✓ System Info | ✓ Notification Options | ✓ Account Management | ✓ Roles & Protocols | ✓ Federation Info | ✖ **Session Timeout** | ✓ Summary

If you are using Single Logout (SLO), the system must keep track of user sessions, which requires some memory. To conserve resources, the system will try to clean up sessions that have been idle for the period of time you specify below.

Session Timeout minute(s) *

To reach this screen:

1. Click **Server Settings** on the Main Menu.
2. Click **Session Timeout** in the steps list.

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 90). 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 42).

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 154). 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.

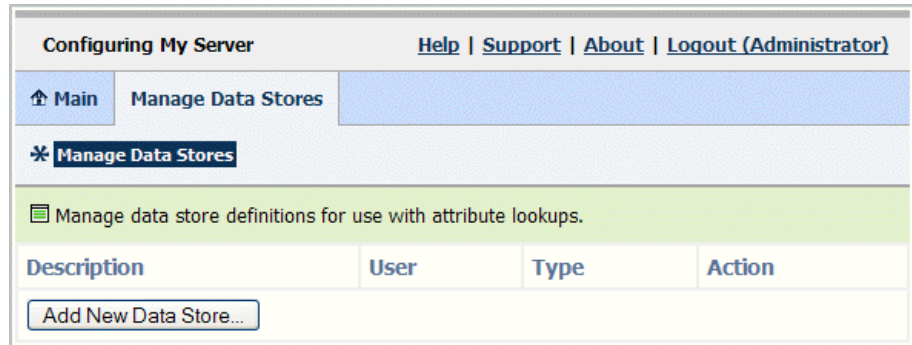
As an SP, you can use data stores to retrieve additional attributes to package with the IdP's assertion data to meet SP adapter requirements (see [“Integration Kits and Adapters”](#) on page 39). Such attributes may be needed, for example, to establish authorization levels or to manage the local account.

You can add data stores at any time. Standard data stores include JDBC-enabled databases and LDAP v3-enabled directory stores. Alternatively, you can develop

a driver using the PingFederate Custom Source SDK to connect to non-JDBC databases (see `README.txt` in the `pingfederate/sdk` directory).



Note: You cannot delete or modify a data store if it is associated with an attribute source as part of a partner-connection configuration. You must remove the association first (see [“Configuring Attribute Sources and User Lookup”](#) on page 154 or [“Selecting an Adapter Data Store”](#) on page 230).



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 79.
 - For LDAP configuration information see [“Configuring an LDAP Connection”](#) on page 82.
 - For Custom configuration information see [“Configuring a Custom Data Store”](#) on page 84.
4. Click **Save** when you return to this screen.

To modify a data store:

- Click the data store Description.
 - For Database configuration information see [“Configuring a JDBC Database Connection”](#) on page 79.
 - For LDAP configuration information see [“Configuring an LDAP Connection”](#) on page 82.

- For Custom configuration information see [“Configuring a Custom Data Store”](#) on page 84.



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.



Note: Ensure that your vendor's driver JAR is installed in the `pingfederate/server/default/lib` directory. You must restart the server after installing the driver.

Configuring My Server

[Help](#) | [Support](#) | [About](#) | [Logout \(Administrator\)](#)

[Main](#)

[Manage Data Stores](#)

[Data Store](#)

Database Config

Summary

Please provide the details for configuring this database connection.

JDBC URL

jdbc:oracle:thin:@//122.16.14.59:152 *

Driver Class

oracle.jdbc.OracleDriver *

Username

admin *

Password

☐ Mask Values in Log

Advanced...

Field Descriptions

Field	Definition
JDBC URL	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.
Driver Class	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.
Username	The name that identifies the user when connecting to the database.
Password	The credentials needed to access the database.
Mask Values in Log (Checkbox)	Determines whether all attribute values returned from this data store will be masked in PingFederate log files (see “Attribute Masking” on page 42).

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 reach this screen for configuring 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 42.

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 81).

6. Click **Next**.



Note: PingFederate will try to connect to the database 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.

Setting Advanced Options

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

Field Descriptions

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

Field	Definition
Validate Connection SQL	(Optional) Initiates a test SQL query to update connections in the pool.

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 reach this screen for configuring 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.

The screenshot shows the 'Configuring My Server' interface. At the top, there are links for 'Help', 'Support', 'About', and 'Logout (Administrator)'. Below this is a navigation bar with 'Main', 'Manage Data Stores', and 'Data Store'. The 'Data Store' section is active, showing 'Data Store Type' as 'LDAP Configuration' and 'Summary'. A green banner instructs the user to 'Please provide the details for configuring this LDAP connection.' The form includes fields for 'Hostname' (required), 'Bind Anonymously' (checkbox), 'Username' (required), 'Password' (masked with asterisks), 'Use SSL' (checkbox), and 'Mask Values in Log' (checkbox).

Field Descriptions

Field	Definition
Hostname	The hostname of the data store.

Field	Definition
Bind Anonymously (Checkbox)	Username and password are not required.
Username	The username credential required to access the data store.
Password	The password credential required to access the data store.
Use SSL (Checkbox)	An option to connect to the LDAP data store using secure SSL/TLS encryption.
Mask Values in Log (Checkbox)	Determines whether all attribute values returned from this data store will be masked in PingFederate log files (see “Attribute Masking” on page 42).

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 reach this screen for configuring a new data store:

1. Click **Data Stores** on the Main Menu.
2. Click **Add New Data Store**.
3. Select LDAP and click **Next**.

To establish a connection to an LDAP data store:

1. Enter the applicable Hostname.
This can be a valid DNS name or IP address.
The Hostname will also be used to identify this data store in lists.
2. 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 Username and Password.



Note: If you choose an anonymous binding, ensure that this access level is sufficient to retrieve the user data you will need for SSO transactions.

3. (Optional) Select Mask Values in Log.
For information see [“Attribute Masking”](#) on page 42.
4. 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.

5. On the Summary screen, click **Done**.
6. Click **Save** on the Manage Data Stores screen.

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 `README.txt` in the `pingfederate/sdk` directory).

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

The screenshot shows a web interface titled "Configuring My Server" with navigation links: [Help](#), [Support](#), [About](#), and [Logout \(Administrator\)](#). Below the title bar is a menu with "Main", "Manage Data Stores", and "Data Store". The "Data Store" menu is active, showing a sub-menu with "Data Store Type", "Custom Data Store Type" (selected), "Configure Attribute Source Adapter Instance", and "Summary". A green instruction box says: "Enter a descriptive name, a system-wide unique ID, and select the custom data store adapter to use." The form contains three fields: "Data Store Instance Name" with the value "DS1" and an asterisk, "Data Store Type" with a dropdown menu showing "- SELECT -" and an asterisk, and a checkbox labeled "Mask Values in Log" which is currently unchecked.

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 reach this screen for configuring a new data store:

1. Click **Data Stores** on the Main Menu.
2. Click **Add New Data Store**.
3. Select Custom and click **Next**.

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 42.

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.

Field Name	Field Value	Description
Primary SOAP Endpoint	<input type="text" value="https://my_company:soap"/>	Primary Endpoint
Secondary SOAP Endpoint	<input type="text" value="https://my_company:soap"/>	Secondary Endpoint
Username	<input type="text" value="admin"/>	Username to allow access to the datastore.
Password	<input type="password" value="....."/>	Password associated to username.

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.
3. Click **Configure Attribute Source Adapter Instance**.

To reach this screen for configuring a new data store:

1. Click **Data Stores** on the Main Menu.
2. Click **Add New Data Store**.
3. Select Custom and click **Next**.
4. Complete information on the Data Store Type screen and click **Next**.

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 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

IdP Discovery provides a cookie-based look-up mechanism that SP application developers can use to identify a user's identity provider (see “[IdP Discovery](#)” on page 32).

The first step in configuring IdP Discovery is to choose the discovery role or roles that PingFederate will play. The choices you see on the 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 74).

The screenshot shows a web interface titled "Configuring My Server" with navigation links for Help, Support, About, and Logout. Below the title bar, there are tabs for "Main", "Configure IdP Discovery", and a third tab. The "Configure IdP Discovery" tab is active. Under this tab, there are three sub-sections: "Domain Cookie Settings" (marked with an asterisk), "Local Common Domain Server" (marked with a checkmark), and "Summary" (marked with a checkmark). The "Domain Cookie Settings" section contains a message: "Please select one or more roles that this service will play in the IdP Discovery profile." Below this message are three checkboxes: "Common domain server that sets and or reads the common domain cookie" (checked), "SP using a common domain service to discover what Identity Providers a principal has recently used" (unchecked), and "IdP using a common domain service to advertise the ability to authenticate a principal" (unchecked).

To reach this screen:

- ▶ Click **IdP Discovery** under System Settings on the Main Menu.

If this link is not available, then you have not yet enabled the IdP Discovery federation role (see “[Choosing Roles and Protocols](#)” on page 74).

For a detailed discussion of selections on this screen, see “[IdP Discovery](#)” on page 32).

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 you have not yet enabled the IdP Discovery federation role (see [“Choosing Roles and Protocols”](#) on page 74).
2. Click the **Configure IdP Discovery** button.
3. Click **Common Domain Service** in the steps list.

This step is not available if your server is configured for IdP Discovery only (see [“Choosing Roles and Protocols”](#) on page 74).

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.

The screenshot shows the 'Configuring My Server' interface. At the top, there are links for 'Help', 'Support', 'About', and 'Logout (Administrator)'. Below this is a navigation bar with 'Main' and 'Configure IdP Discovery'. Under 'Configure IdP Discovery', there are three options: 'Domain Cookie Settings' (checked), 'Common Domain Service' (checked), and 'Local Common Domain Server' (selected with a star icon). Below the navigation bar, a green box contains the instruction: 'Please provide configuration information for this server to act as the common domain service.' The configuration form has four fields: 'Common Domain' with the value '.pingidentity.com' and a required field asterisk; 'Cookie Lifetime (days)' with the value '1000' and a required field asterisk; 'Pass phrase' with a masked input (seven dots); and 'Confirm Pass' with a masked input (seven dots).

To reach this screen:

1. Click **IdP Discovery** under System Settings on the Main Menu.
If this link is not available, then you have not yet enabled the IdP Discovery federation role (see “[Choosing Roles and Protocols](#)” on page 74).
2. Click the **Configure IdP Discovery** button.
3. Click **Local Common Domain Server** in the steps list.
This step is available only if you have selected the common-server option under Domain Cookie Settings (see “[Configuring IdP Discovery](#)” on page 86).

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.
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.

System Administration

This chapter describes general administrative functions for PingFederate, including:

- [“Starting and Stopping PingFederate”](#) on page 90
- [“Log File Generation”](#) on page 90
- [“Exporting Metadata”](#) on page 94
- [“Signing XML Files”](#) on page 97
- [“Using the Configuration Archive”](#) on page 99
- [“Managing Email Configuration”](#) on page 100
- [“Account Management”](#) on page 101
- [“Using Virtual Host Names”](#) on page 105
- [“Installing a New License Key”](#) on page 106
- [“Changing Configuration Parameters”](#) on page 107
- [“Using Velocity Templates”](#) on page 110



Note: The information in this chapter is presented from the viewpoint of an administrative user with “Admin” permissions (see [“Account Management”](#) on page 101).

Starting and Stopping PingFederate

(Windows)

To start PingFederate:

- From Start > Run dialog or a command prompt, run the batch file:
`<PF_install_dir>\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]s:[yy]ms”.

To shut down PingFederate:

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

(Linux)

To start PingFederate:

1. From a command prompt, change directories to `<PF_install_dir>/pingfederate/bin`.
2. Add executable permission to the startup script:
`chmod 755 ./run.sh`
3. 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]s:[yy]ms”.

To shut down PingFederate:

- Enter Ctrl+C in the terminal window.

Log File Generation

PingFederate generates log files that document the system's activities. These logs are stored in the `pingfederate/log` directory and include:

- `admin.log` — Records all actions performed by administrative console users (see [“Administrator Audit Logging”](#) on page 91)
- `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 92).

- `server.log` — Records all PingFederate runtime and administrative server activity

A JBoss-generated log file, `boot.log`, is also located in this directory.



Tip: PingFederate logs user attributes, when they 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 42).

Logging is 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>).

The following sections provide more detail about the `admin` and `transaction` logs (see [“Administrator Audit Logging”](#) on page 91 and [“Runtime Transaction Logging”](#) on page 92).

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` and the `admin.log` file 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 92); 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.)

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 of the “Managing Connections: . . .” chapters).

You also have the option of overriding transaction logging for all connections (to find this feature, click the relevant **Manage All . . .** 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 3: Transaction Logging Modes

Mode	Description
None	No transaction logging.
Standard	<p>(Default) Logs summary information for each transaction message, including:</p> <ul style="list-style-type: none"> • Timestamp • Hostname:Port • LogMode • ConnectionID • SAML Status Code> (for SAML responses only) • Context • MessageType • SAML ID (for SAML messages only) • Endpoint (for outbound messages only) • Target URL (if SSO transaction)
Enhanced	<p>Includes everything logged at the Standard level plus:</p> <ul style="list-style-type: none"> • SAML_SUBJECT* • Binding • RelayState (if available) • SignaturePolicy • SignatureStatus • HTTP Request Parameters (outbound messages only) <p>* Only when available in a SAML assertion, a single-logout request, a Request Security Token Response (RSTR), or an authentication request (AuthnRequest)</p>

Table 3: Transaction Logging Modes (Continued)

Mode	Description
Full	Includes everything logged at the Enhanced level plus the complete XML message for every transaction.

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 deployments. You export metadata via the Main Menu. You can import your partner's metadata file, when available, at the beginning of the connection-configuration process (see [“Configuring IdP Connections”](#) on page 211 or [“Configuring SP Connections”](#) on page 132).



Note: After exporting a metadata file, you can choose to digitally sign the file before sending it to your connection partner (see [“Signing XML Files”](#) on page 97).

To reach this screen:

- Click **SAML Metadata Export** 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 **Use a Connection** and click **Next**.
3. On the Export Connection Metadata screen, select the connection from the drop-down menu and click **Next**.
4. On the Export Metadata 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 choose **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” next.
 - “Choosing a Metadata Signing Key” on page 96.
 - “Export XML Encryption Certificates” on page 96.
5. On the Export Metadata 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 41.

Configuring My Server		Help Support About Logout (Administrator)
Main	Export Metadata	
✓ Metadata Role	✓ Metadata Mode	✳ Attribute Contract
✓ Protocol	✓ Signing Key	✓ Export XML Encryption Certificate
<p>Define the Attribute Contract you want to include in this metadata file. Note that if you do not specify any attributes, no contract will be included in the generated metadata file.</p>		
		Action
		Add

To reach this screen:

1. Click **Export Metadata** on the Main Menu.
2. On the Metadata Mode screen, click the radio button for selecting 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 your 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 118.

The screenshot shows a web interface titled "Configuring My Server" with navigation links for Help, Support, About, and Logout (Administrator). The "Export Metadata" tab is active, showing a progress bar with several steps: Metadata Role, Metadata Mode, Protocol, Attribute Contract, Signing Key (highlighted with a blue box and a star icon), and Export XML Encryption Certificate. Below the progress bar, a green box contains a message: "The metadata may contain a public key that this system uses for digital signatures. If you wish to include a key, please select from the list of available signature keys." Underneath this message is a section titled "Digital Signature Keys/Certs" containing a drop-down menu currently set to "- SELECT -".

To export your public signature verification key:

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

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



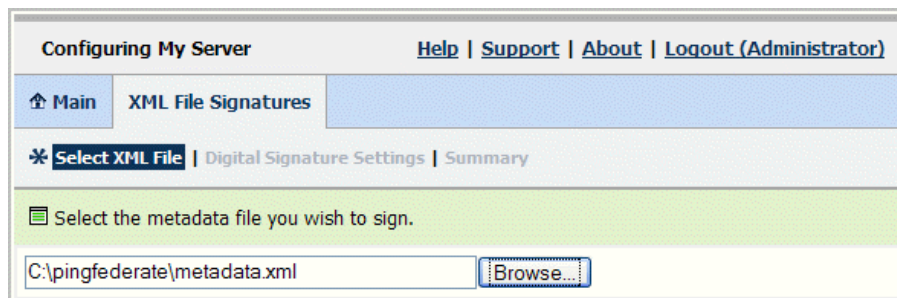
Important: To finish the download, you must click the **Export** button at the bottom left of the Export Metadata screen.

Signing XML Files

PingFederate supports digital signing of SAML [metadata](#) files or any other XML files that you and your partner may 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 216.
- As an IdP configuring an SP connection, see [“Importing Metadata”](#) on page 138.
- ▶ XML file signing is available from the Main Menu under Administrative Functions.



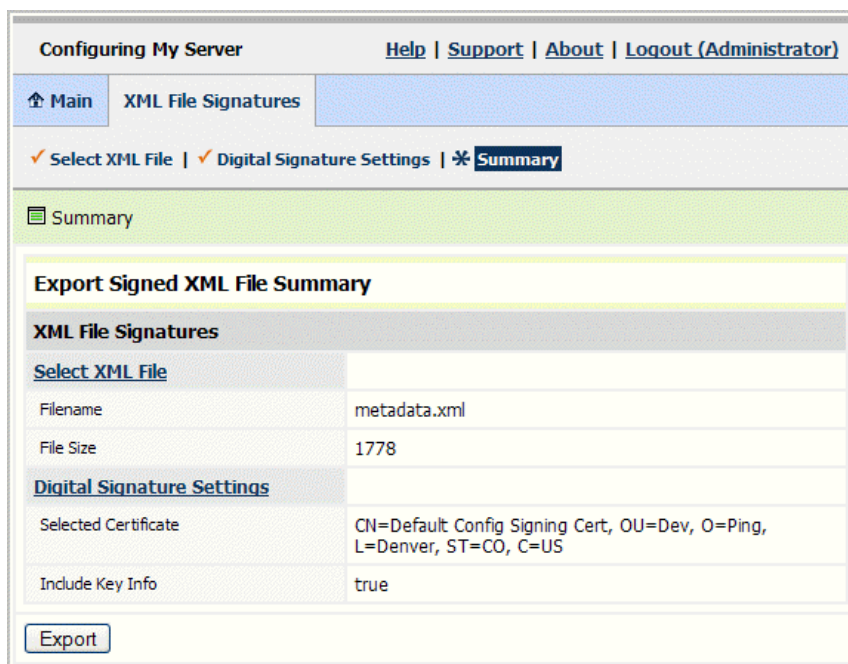
To sign an XML file:

1. On the Select XML File screen, locate and open the file.
2. On the Select Signing Key screen, choose the certificate containing your signing key from the drop-down list.



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

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



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

Using the Configuration Archive

PingFederate's archive utility allows you to download your configuration to a zip file. You can use this file to restore server configurations. A configuration archive is created automatically every time you log on to the administrative console. The archives are stored in `pingfederate/server/default/data/archive`.

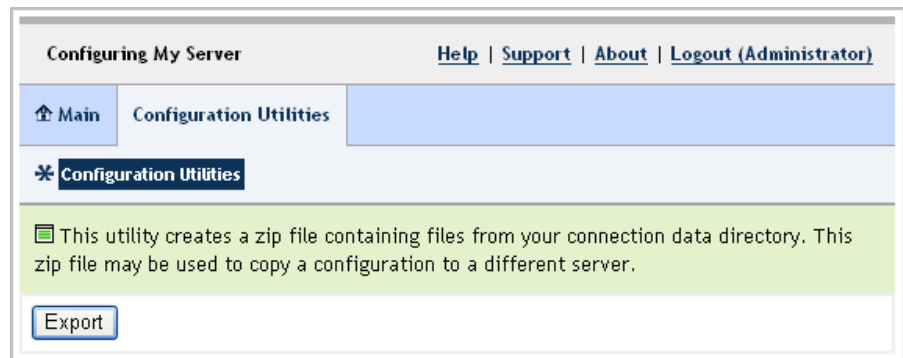
Configuration archives can also be used to transfer data to a new point release of PingFederate—for example, to upgrade from version 4.0 to 4.3.



Note: The configuration archive does not include error-page or other end-user HTML templates (see [“Using Velocity Templates”](#) on page 110). If any changes have been made to these pages, you must copy them over to new installations of PingFederate.



Caution: Configuration archives used for new releases must not contain any draft connections. Either complete or remove any unfinished partner connections in your current PingFederate version before creating the upgrade archive.



To reach this screen:

- Click **Configuration Archive** on the Main Menu.

To save an archive:

- On the Configuration Utilities screen, click **Export** and save the download to your file system, then click **Done**.

To deploy an archive:

1. Copy the file into the directory:
`<PF_install_dir>/pingfederate/server/default/data/drop-in-deployer`

2. Rename the copied file to `data.zip`.

When the PingFederate server is running, the file disappears after a moment and the data automatically deploys.



Caution: A deployed archive overwrites any existing configuration data.

Managing Email Configuration

If you are using the email notification option for either user passwords or licensing events, you must set up and maintain a connection to the email server that PingFederate will use to send messages (see “[Configuring Notification Options](#)” on page 71).

Field Descriptions

Field	Definition
“From” Address	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.
Email Server	The IP address or hostname of your email server.

Field	Definition
SMTP Port	The SMTP port on your email server (default: 25).
Username	Authorized email username.
Password	User password.
Confirm Password	Re-entered password.
Test Address	An email address where you want to send a test message.

To reach this screen:

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

If this link is not showing, then email notification is not configured in Server Settings (see [“Configuring Notification Options”](#) on page 71).

To configure access to your email server:

1. Enter information into all fields. (Username and Password are not required.)
2. Optionally, 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 receives a message from the server.

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

Account Management

PingFederate provides a choice of single- or multi-user system administration (see [“Setting Administration Style”](#) on page 70). If you choose multi-user

administration, the system provides role-based access control, as shown in [Table 4](#).

Table 4: PingFederate User Access Control

Role Assignment	Access Privileges
User Admin	Add new users to the system, change permissions, and reset passwords for existing users. Also select administration style (single- or multi-user), define email notification policies, and configure SMTP server connection.
Admin	Configure partner connections and most system settings (except user management and certificate handling)
Crypto Admin	Manage certificates.
Auditor	View-only permission for all Admin functions.

When Auditor is assigned, no other roles may be set. Admin users may have multiple roles set.

To reach the Account Management screen:

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

User Name	User Admin	Admin	Crypto Admin	Action
Administrator	<input type="radio"/> Auditor <input checked="" type="radio"/> Admin	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Deactivate / Change Password

Create User

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 only change their own passwords from this screen (see [“Changing Passwords”](#) on page 105).

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 have elected to have PingFederate send passwords via email, in which case you must supply an email address (see [“Configuring Notification Options”](#) on page 71).

3. Optionally, enter additional information.
4. Click **Next** to set up a password (see the next section).



Note: 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 4](#) on page 102).

Clicking the Auditor button deactivates other permission selections.

To deactivate a user:

- ▶ Click **Deactivate** for the user.



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

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).

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

The screenshot shows a web application titled "Configuring My Server". At the top right are links for [Help](#), [Support](#), [About](#), and [Logout](#). Below this is a navigation bar with tabs: [Main](#), [Account Management](#), and [User Information](#). The [User Information](#) tab is active, and within it, the [Password Generation](#) sub-tab is selected, indicated by a star icon. Below the navigation bar, there are three status indicators: a checkmark for "User Information", a star for "Password Generation", and a checkmark for "Summary". A green message box states: "This system generated password is valid for only a single use. Upon subsequent login, the user will be required to change his/her password." Below this message is a button labeled "Generate one-time password". Underneath the button is a text box containing the generated password "wfQ37YnV".

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 this feature—see [“Configuring Notification Options”](#) on page 71).

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.

The screenshot shows a web interface titled "Configuring My Server". At the top right are links for "Help", "Support", "About", and "Logout (Administrator)". Below this is a navigation bar with "Main", "Account Management", and "Change Password". The "Change Password" section is active, showing a message: "Please provide matching New and Confirm Passwords." Below this is a form with four fields: "Username" (pre-filled with "Administrator"), "Current User Password", "New Password", and "Confirm New Password". Each password field has a text input box.

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 (twice) 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.

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 47). Depending on your needs, however, you can configure virtual server IDs and virtual hosts in the same installation of PingFederate.

Configuring My Server	
Help Support About Logout (Administrator)	
Main Manage Virtual Host Names	
* Virtual Host Names	
<div>Optionally, you can define a list of alternate domain names at which this server may receive XML protocol messages. This option allows more flexibility for validating protocol elements such as <Recipient> and <Destination> in inbound messages.</div>	
Host Domain Name	Action
<input type="text"/>	Add

Installing a New License Key

If your license expires, you must install a new license key.



Note: If you are using PingFederate’s initial free license, the server will shut down after ninety days. You can configure the server to send an email in advance (see “[Configuring Notification Options](#)” on page 71).

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

When your license key has expired:

1. Find your server ID and product version number on the administrative console’s **About** page.

If your console is not running, you may use the command line (see “[Locating Your Server ID](#)” on page 107).

2. Request a license key from Ping Identity.

You will need your server ID and PingFederate version number. For more information, see “[Installing PingFederate](#)” on page 54.

3. You will receive the license key via email.
4. Save the `pingfederate.lic` file to `pingfederate/server/default/conf`.



Important: The license key *must* be named `pingfederate.lic`. It may take a minute for a running server to recognize the new key.

Locating Your Server ID

Your server ID is a unique identifier generated by PingFederate. Use this ID, with the accompanying product version number, for technical support questions or to obtain a new license key (see [“Installing a New License Key”](#) on page 106).

You can find the server ID by clicking the **About** link in the top navigation bar of the administrative console or looking in this text file:

```
pingfederate/bin/pingfederate-server-id.txt
```

Changing Configuration Parameters

PingFederate’s runtime parameters are contained in the file `run.properties`, located in: `<PF_install_dir>/pingfederate/bin`. [Table 5](#) describes the properties; refer to the file itself for default settings not specified here. You can change these settings as needed.

Table 5: PingFederate Runtime Properties

Property	Description
<code>pf.admin.https.port</code>	Defines the port on which the PingFederate administrative console runs. Default is 9999.
<code>pf.console.bind.address</code>	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.
<code>pf.http.port</code>	Defines the port on which PingFederate listens for unencrypted HTTP traffic. Default is 9030.
<code>pf.https.port</code>	Defines the port on which PingFederate listens for encrypted HTTPS (SSL/TLS) traffic. Default is 9031.

Table 5: PingFederate Runtime Properties (Continued)

Property	Description
pf.secondary.https.port	<p>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.*</p> <p>Additional configuration of the listener ports (including adding new listeners) is available via the <code>/server/default/deploy/jetty.sar/META-INF/jboss-service.xml</code> file. Of particular value are the <code>WantClientAuth</code> and <code>NeedClientAuth</code> flags, which indicate to a client the request or requirement, respectively, for a client certificate. These are set to false by default.</p>
pf.engine.bind.address	<p>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.</p>
pf.ajp.port	<p>Defines the listening port for the Apache JServ Protocol (AJP) v13 protocol, for use with a proxy server. To use this setting, you must uncomment the AJP section in <code>server/default/deploy/jetty.sar/META-INF/jboss-service.xml</code>.</p>
pf.ajp.confidential.port	<p>Defines the confidential/encrypted AJP listening port.</p>
pf.operational.mode	<p>Controls the operational mode of the PingFederate server. Refer to the properties file for a list of valid values. (See “Clustering and Failover Deployment” on page 295.)</p>
pf.cluster.multicast.ip	<p>Defines the IP address shared among nodes in the same cluster for multicast communication to keep failover instances of PingFederate synchronized. Note: nodes in a cluster must use the same address for this property. (See “Clustering and Failover Deployment” on page 295.)</p>
pf.subcluster.cluster.multicast.ip	<p>Defines the IP address shared among nodes in the same subcluster for multicast communication. Note: nodes in a subcluster must use the same address. (See “Clustering and Failover Deployment” on page 295.)</p>

Table 5: PingFederate Runtime Properties (Continued)

Property	Description
pf.secondary.https.port	<p>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.*</p> <p>Additional configuration of the listener ports (including adding new listeners) is available via the <code>/server/default/deploy/jetty.sar/META-INF/jboss-service.xml</code> file. Of particular value are the <code>WantClientAuth</code> and <code>NeedClientAuth</code> flags, which indicate to a client the request or requirement, respectively, for a client certificate. These are set to false by default.</p>
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.ajp.port	Defines the listening port for the Apache JServ Protocol (AJP) v13 protocol, for use with a proxy server. To use this setting, you must uncomment the AJP section in <code>server/default/deploy/jetty.sar/META-INF/jboss-service.xml</code> .
pf.ajp.confidential.port	Defines the confidential/encrypted AJP listening port.
pf.operational.mode	Controls the operational mode of the PingFederate server. Refer to the properties file for a list of valid values. (See “Clustering and Failover Deployment” on page 295.)
pf.cluster.multicast.ip	Defines the IP address shared among nodes in the same cluster for multicast communication to keep failover instances of PingFederate synchronized. Note: nodes in a cluster must use the same address for this property. (See “Clustering and Failover Deployment” on page 295.)
pf.subcluster.cluster.multicast.ip	Defines the IP address shared among nodes in the same subcluster for multicast communication. Note: nodes in a subcluster must use the same address. (See “Clustering and Failover Deployment” on page 295.)

Table 5: PingFederate Runtime Properties (Continued)

Property	Description
pf.cluster.bind.address	Defines the IP address of a particular network adapter over which multicast communication occurs. Use for deployments where multiple network interfaces are installed on the machine running PingFederate. If blank (the default), the PingFederate server defaults to the first non-loopback address it finds.
pf.subcluster.bind.address	Defines the IP address of a particular network adapter over which multicast communication occurs for a subcluster. Use for deployments where multiple network interfaces are installed on the machine running PingFederate. If blank (the default), the PingFederate server defaults to the first non-loopback address it finds.

Using Velocity Templates

PingFederate supplies Web-page templates for error conditions and other situations that an end-user might encounter during an SSO or SLO transaction. These default pages utilize the Velocity template engine, an open-source Apache project, and are located in the `<PF_install_dir>/pingfederate/server/default/conf/template` directory. The filenames indicate the purpose of each page.

At runtime, the user's browser is directed to the appropriate page, depending on the operation being performed and where the error occurs. 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.

You can modify 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 first interprets the commands embedded within the template landing page and then renders the HTML in the user's browser. Variable information is supplied by the PingFederate runtime engine.

For information on how to modify the Velocity functions, please refer to the Velocity project documentation on the Apache Web site:

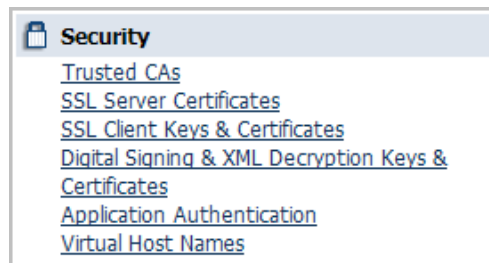
<http://velocity.apache.org/engine/releases/velocity-1.4>

Applications can override any of the default pages provided specifically for SSO and SLO errors by specifying a URL value in the relevant endpoint's `InErrorResource` parameter. For more information, refer to “[Application Endpoints](#)” on page 289.

Security Management

PingFederate provides built-in certificate management to handle security considerations for SAML transactions. In addition, when you use the SAML 2.0 XASP profile as an SP, password security is required between the application requesting attributes and the SP PingFederate server.

You configure these features via the Security section under My Server on the Main Menu.



Note: The information in this chapter is presented from the viewpoint of an administrative user with "Crypto Admin" permissions (see ["Account Management"](#) on page 101).

Trusted CAs

You can import your federation partner's Certificate Authority (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.



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:

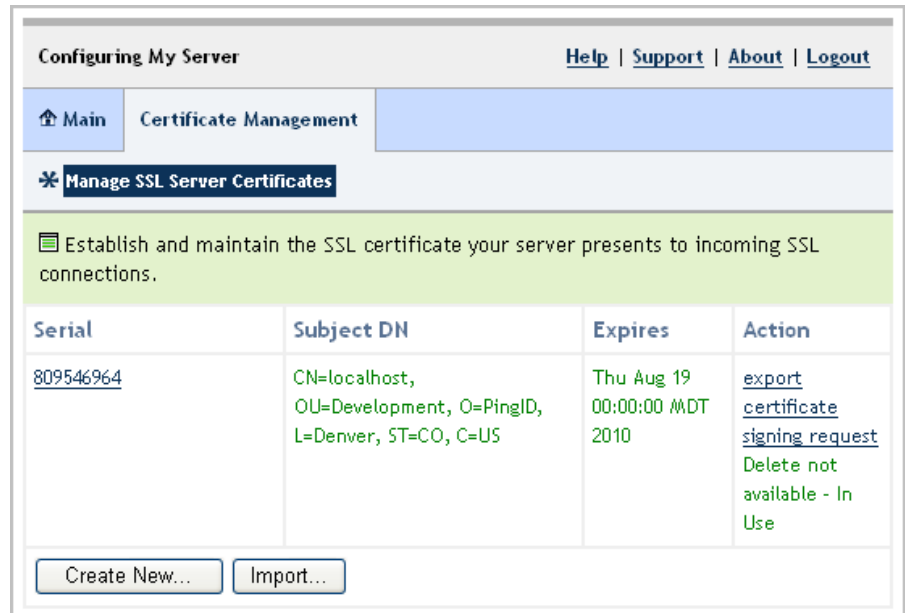
1. Click **Delete** under Action for the certificate you want to delete.
To undo the deletion, click **Undelete**.
2. Click **Save**.

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 that your PingFederate server presents to any incoming SSL/TLS connections.



To reach this screen:

- Click **SSL Server Certificates** on the Main Menu.

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.

To activate a certificate:

- 1 Click **Activate** under Action for the certificate you want to activate.

This function is available only if you have more than one certificate. It will appear for any inactive certificates. You can have only one active SSL server certificate.

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 signature request:

1. Click **Certificate Signing Request** under Action for the certificate for which you want to generate a request.
2. Select Generate Certificate Signing Request (CSR), if not already selected.
3. Click **Next**.
4. On the Certificate Summary screen, click **Export**.
5. Save the file on your system and click **Done**.

To import a certificate authority response:

1. Click **Certificate Signing Request** under Action for the certificate for which you want to import a response.
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**.
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**.

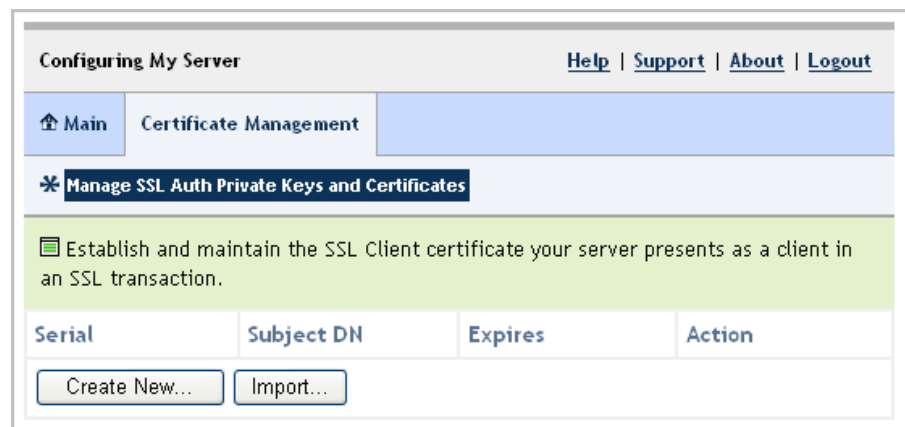
2. Click **Save**.

Create Certificate Field Descriptions

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

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**.
4. On the Certificate Summary screen, click **Export**.
5. Save the file on your system and click **Done**.

To create a certificate authority signature request:

1. Click **Certificate Signing Request** under Action for the certificate for which you want to generate a request.
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 Request** under Action for the certificate for which you want to import a response.
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:

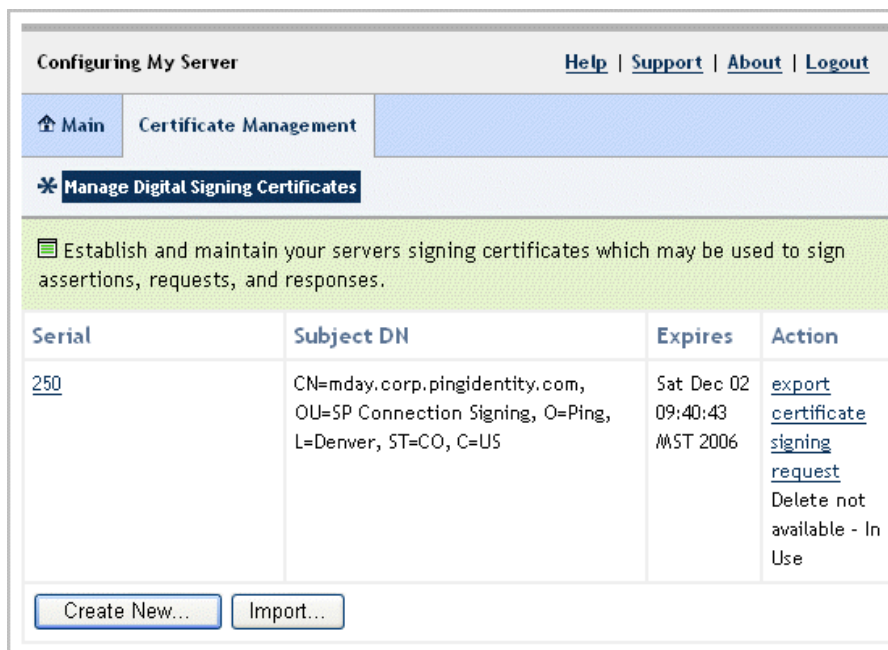
1. Click **Delete** under Action for the certificate you want to delete.
(To undo the deletion, click **Undelete**.)
2. Click **Save**.

Create Certificate Field Descriptions

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

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 46).



To reach this screen:

- Click **Digital Signing 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.

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**.
4. On the Certificate Summary screen, click **Export**.
5. Save the file on your system and click **Done**.

To create a certificate authority signature request:

1. Click **Certificate Signing Request** under Action for the certificate for which you want to generate a request.
2. Select Generate Certificate Signing Request (CSR), if not already selected and click **Next**.
3. Click **Export**.
4. Save the file on your system and click **Done**.

To import a certificate authority response:

1. Click **Certificate Signing Request** under Action for the certificate for which you want to import a response.
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**.

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

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

Application Authentication

If you are using the SAML 2.0 XASP profile as an SP, then the requesting application(s) at your site must authenticate to your PingFederate server (see [“Attribute Query and XASP”](#) on page 31). You configure this authentication on the Application Authentication screen. For application information see [“/sp/startAttributeQuery.ping”](#) on page 294.

In addition, you may want to require IdP or SP applications to authenticate if they call PingFederate’s SSO Directory Service (see [“SSO Directory Service”](#) on page 303).

Configuring My Server [Help](#) | [Support](#) | [About](#) | [Logout \(Administrator\)](#)

[Main](#) > Manage Application Authentication

* Application Authentication

Applications requesting user attributes must authenticate using a shared secret. You might also want to require that applications authenticate in order to access PingFederate's built-in Web services. For each application, provide a unique ID and corresponding shared secret. Click the checkbox to require Web services authentication.

Application Id	Shared Secret	Confirm Shared Secret	Action
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="button" value="Add"/>

☐ Require HTTP Basic authentication for applications connecting to the SSO directory Web services

To configure application authentication:

1. Enter an Application Id and Shared Secret in their respective text boxes.
You and the application developer must agree to these values.
2. Enter confirmation of the Shared Secret and click **Add**.
3. Repeat the steps above for any other applications, as needed.
4. If authentication is required for an application to use PingFederate Web services, then click the indicated checkbox.



Tip: To block access to the PingFederate Web Directory Service for security purposes, you can select the checkbox without entering any authentication values, or you can remove the deployment WAR file and restart the server, if needed (see [“SSO Directory Service”](#) on page 303).

To change an application ID or password:

1. Delete the existing line and reenter new information.
2. If you are editing an existing configuration, click **Save** to confirm the change.

Identity Provider Configuration

In an IdP role, you manage connections to SAML or WS-Federation gateways located at 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 74).



Note: With PingFederate you can operate your enterprise in either an IdP, SP, or IdP Discovery role, or in any combination of roles (see [“Managing Server Settings”](#) on page 69).

Note that you configure only one connection per federation partner, even if you are targeting more than one Web application at the destination SP site using the same federation protocol.

The integration of applications with PingFederate is a critical aspect of providing end-users with access to services across domains. This process is facilitated through the use of application integration kits and a robust Software Development Kit (see [“Integration Kits and Adapters”](#) on page 39).

This chapter covers the following topics:

- [“Application Integration Settings”](#) on page 124
- [“Viewing Protocol Endpoints”](#) on page 131
- [“Configuring SP Connections”](#) on page 132
- [“Defining SP Affiliations”](#) on page 193

Application Integration Settings

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

Configuring IdP Adapters

An IdP adapter is used to look up session information and provide user identification to PingFederate (see [“Integration Kits and Adapters”](#) on page 39).

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 standard or the LDAP adapter for the first time as part of the post-installation process, see [“Configuring the IdP Standard Adapter”](#) on page 269 or [“Configuring the IdP LDAP Adapter”](#) on page 281, respectively.

Configuring My Server

[Help](#) | [Support](#) | [About](#) | [Logout \(Administrator\)](#)

Main

Manage IdP Adapter Instances

✱ Manage Adapter Instances

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.

Adapter Instance Id	Adapter Instance Name	Adapter Type	Action
idpadapter	idpadapter	PF4 Standard Adapter v1.2	None Available - In Use

Create New Adapter Instance...

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

To configure a new adapter instance:

- ▶ Click **Create New Adapter Instance**.

To edit an existing adapter instance:

- ▶ Click the Adapter Instance Name and click the step you need to change.

To delete an adapter instance:

1. Click **Delete** next to the Adapter 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 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_dir>/server/default/deploy directory. Some adapters are bundled with PingFederate (see “[Integration Kits and Adapters](#)” on page 39). Other adapters and integration kits are available from the [Ping Identity Web site](http://www.pingidentity.com) (www.pingidentity.com).

Field Descriptions

Field	Definition
Adapter Instance Name	A descriptive name for the adapter instance—for example, an identity management system name.
Adapter Instance ID	An internal identifier for the adapter instance. Must be alphanumeric with no spaces.
Drop-down list	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.

To define an adapter instance:

1. Enter the Adapter Instance Name and Adapter Instance Id on the Adapter Type screen.
2. Select the Adapter Type from the drop-down menu.

If the adapter you need is not listed, click **Visit PingIdentity.com for Additional Adapters** to see if a suitable adapter is available from the PingFederate download site, or create your own adapter (see [“Integration Kits and Adapters”](#) on page 39).

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 [“Integration Kits and Adapters”](#) on page 39). For information about configuring the adapters packaged with PingFederate, see [“Standard Adapter Configuration”](#) on page 267 or [“LDAP Adapter Configuration”](#) on page 279.

4. Click **Done** on the Adapter Summary screen.
5. Click **Save** on the Manage 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 [“Integration Kits and Adapters”](#) on page 39).

- ▶ For information about configuring the Standard Adapter, see [“Standard Adapter Configuration”](#) on page 267.
- ▶ For information about configuring the LDAP Authentication Service, see [“LDAP Adapter Configuration”](#) on page 279.



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 beside the adapter selection drop-down. 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 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 Standard Adapter, see [“Configuring the IdP Standard Adapter”](#) on page 269.
- For information about actions available using the LDAP Authentication Service, see [“Configuring the IdP LDAP Adapter”](#) on page 281.

Action Name	Action Description	Action Invocation Link
Generate properties	Generate properties for the agent side	Invoke Generate properties

To reach this screen:

1. Click **Adapters** on the Main Menu.
2. Click an Adapter Instance Name on the Manage Adapter Instances screen.
3. Click **Adapter Actions** (if available).

To generate a properties list:

- Click **Generate properties** 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 Standard Adapter, for example, provides such an option (see [“Standard Adapter Configuration”](#) on page 267).

Configuring IdP Adapter

[Help](#) | [Support](#) | [About](#) | [Logout \(Administrator\)](#)

⬆ Main

Manage IdP Adapter Instances

Create Adapter Instance

✓ Adapter Type | ✓ IdP Adapter | ✓ Adapter Actions | ✖ **Extended Adapter Contract** | ✓ Adapter Attributes | ✓ Summary

📖

This adapter type supports the creation of an Extended Adapter Contract after initial deployment of the adapter instance. This Adapter Contract may be used to fulfill the Attribute Contract, look up additional attributes from a local data store, or create a persistent name identifier which uniquely identifies the user passed to your SP partners.

Adapter Contract

userId

Extend the Contract

email

Action

[Edit](#) / [Delete](#)

Add

To reach this screen:

1. Click **Adapters** on the Main Menu.
2. Click an Adapter Instance Name on the Manage Adapter Instances screen.
3. Click **Extended Adapter Contract** (if available).

To add an attribute:

- Enter the attribute name in the text box and click **Add**.

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 38).

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 42).

Configuring IdP Adapter
[Help](#) | [Support](#) | [About](#) | [Logout \(Administrator\)](#)

Main
Manage IdP Adapter Instances
Create Adapter Instance

✓ Adapter Type | ✓ IdP Adapter | ✓ Adapter Actions | ✓ Extended Adapter Contract | ✖ Adapter Attributes | Summary

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.

Attribute	Pseudonym	Mask Log Values
userId	<input type="checkbox"/>	<input type="checkbox"/>

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.

To reach this screen:

1. Click **Adapters** on the Main Menu.
2. Click an Adapter Instance Name on the Manage Adapter Instances screen.
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](#)” on page 15).

For background information, see the OASIS SAML document: [saml-authn-context-2.0-os](#).

- To enter an authentication context URN for an adapter that supports this feature, click **Advanced Fields** and find the Authentication Context Value field.

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 Adapter Instances screen.

To save an adapter instance:

1. Click **Done** on the Summary screen.
2. Click **Save** (or **Next**) on the Manage 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; 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 will provide built-in destination for the user. This Web page is among the templates you can modify with your own branding or other information (see [“Using Velocity Templates”](#) on page 110).

Configuring My Server [Help](#) | [Support](#) | [About](#) | [Logout \(Administrator\)](#)

[Main](#) **IdP Default URL**

*** IdP Default URL**

Enter values that affect the user's experience when executing IdP-initiated Web SSO operations.

Provide the default URL you would like to send the user to when Single Logout (SLO) has succeeded.

*

Provide the error text displayed in a user's browser when an SSO operation fails.

*

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 may need to know the application endpoints to initiate transactions via PingFederate (see [“Integration Kits and Adapters”](#) on page 39).



Note: For specific parameters required or allowed for Application Endpoints, see [“Application Endpoints”](#) on page 289.

Viewing Protocol Endpoints

Click Protocol Endpoints under Federation Settings in the IdP Configuration section of the Main Menu to see a list of SAML and/or WS-Federation endpoints—a pop-up window displays only those endpoints related to the federation protocols you have chosen (see [“Choosing Roles and Protocols”](#) on page 74). These endpoints are built into PingFederate and cannot be changed.

Your federation partners need to know the applicable IdP Services endpoints to communicate with your PingFederate server. Configured service endpoints are included in metadata export files (see [“Exporting Metadata”](#) on page 94).

The table below describes each endpoint:

Table 6: PingFederate IdP Endpoints

Service	URL and Description
Single Logout Service (SAML 2.0)	<code>/idp/SLO.saml2</code> The URL that receives and processes logout requests and responses.
Single Sign-on Service (SAML 2.0)	<code>/idp/SSO.saml2</code> The SAML 2.0 implementation URL that receives authentication requests for processing.
Artifact Resolution Service (SAML 2.0)	<code>/idp/ARS.ssaml2</code> The SOAP endpoint that processes artifacts returned from a federation partner to retrieve the referenced XML message on the back channel.
Attribute Query Service (SAML 2.0)	<code>/idp/attrsvc.ssaml2</code> The SAML implementation that receives and processes attribute requests.
Single Sign-on Service (SAML 1.x)	<code>/idp/isx.saml1</code> The SAML 1.x implementation of IdP intersite transfer service (ISX) to which clients are redirected for SSO requests.
Artifact Resolution Service (SAML 1.x)	<code>/idp/soap.ssaml1</code> The SOAP endpoint that processes artifacts returned from a federation partner to retrieve the referenced XML message on the back channel.
Single Sign-on Service (WS-Federation)	<code>/idp/prp.wsf</code> The WS-Federation implementation URL that receives and processes security-token requests and SLO messages.

Configuring SP Connections

As an IdP, you configure connection settings to support the exchange of federation protocol messages (SAML or WS-Federation) with an SP. These settings include:

- Attributes you expect to send in an SSO assertion (if any) or 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 [“Standards Support”](#) on page 13.

To continue with the configuration, you and your federation partner must have decided this information in advance (see [“Federation Planning Checklist”](#) on page 46). Your federation partner must supply some connection settings and other information (see [“Configuration Data Exchange”](#) on page 49).

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 77).

Accessing Connections

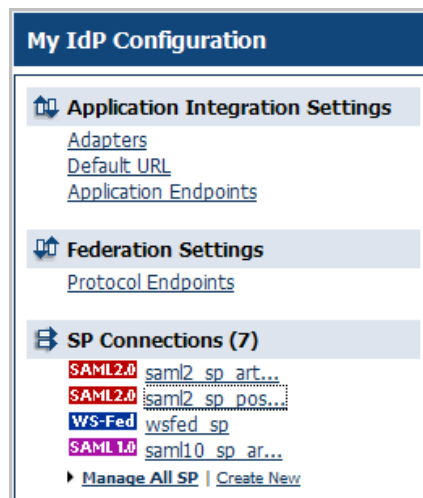
You can create or modify connections directly via the Main Menu. Note that the menu displays only four connections, listed under SP Connections in order according to the most-recently modified. To view a list of all SP connections, click the **Manage All SP** link.

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 recover connection drafts, click **Manage All SP** (see [“Using the Connection List Screen”](#) on page 134).



Note that long connection names are truncated for this display. The full connection names are displayed on the Select a Connection screen (see [“Using the Connection List Screen”](#) on page 134).

To begin configuring a new connection:

- ▶ Click **Create New** under SP Connections on the Main Menu.
See [“Configuration Steps”](#) on page 136 for step-by-step information.



Tip: If you want to use a virtual ID for a second a connection to the same partner, the fastest way is to click **Manage All SP** and copy the first connection (see [“Using the Connection List Screen”](#) on page 134). For information about virtual IDs, see [“Federation Server Identification”](#) on page 47.

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 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 Connection List Screen

From the Select a Connection screen, you can configure a new connection, modify or copy an existing connection or draft, or delete a connection (if it is not active). You can also globally override transaction logging levels set for individual connections or restore connection-based logging (see [“Runtime Transaction Logging”](#) on page 92).

Manage Connections [Help](#) | [Support](#) | [About](#) | [Logout \(Administrator\)](#)

[Main](#) **Manage SP Connections**

Select a Connection

On this screen you can manage connections to your partner SPs. Use the dropdowns to filter the connection list. You can also override the logging mode for all SP connections by specifying a single, global logging mode.

Connection ID	Virtual ID	Protocol	Status	Action
saml10		SAML 1.0	Active	In Use / Copy
saml11		SAML 1.1	Active	In Use / Copy
saml2		SAML 2.0	Active	In Use / Copy
wsfed		WS-Fed	Active	In Use / Copy

[Create Connection...](#)

Logging Mode Override
☒ Off
☐ On

To reach the Select a Connection screen:

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

To begin configuring a new SP connection:

- Click **Create Connection** on the Select a Connection screen.

See “[Configuring SP Connections](#)” on page 132 for step-by-step information.



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 47.

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 139).
3. Make further changes needed for the new connection.

To edit a connection or continue working on a draft:

- Click the Connection ID link.

For a draft, you will return to where you left off.

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.

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 “[Standards Support](#)” on page 13). Also, for any given connection, some configuration steps are required under the applicable protocol, while others are optional. The PingFederate administrative console determines the required steps and dynamically presents optional steps based on your selections.

The remainder of this section provides sequential information about every step you might encounter, 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 discussion always apply to the other protocols, unless otherwise indicated.

Use the lists and links (or page references) below to find specific information about steps that may apply to your connection requirements:

SAML 2.0 Configuration Steps

- [“Selecting a Protocol”](#) on page 138
- [“Importing Metadata”](#) on page 138
- [“General Information”](#) on page 139
- [“Setting an Assertion Lifetime”](#) on page 141
- [“Choosing SAML Profiles”](#) on page 141
- [“Configuring Web SSO”](#) on page 143
- [“Configuring the Attribute Query Profile”](#) on page 180



Note: You can configure the Attribute Query profile as a stand-alone connection or in conjunction with a SAML 2.0 Web SSO connection.

- [“Configuring Credentials”](#) on page 184
- [“Connection Activation and Summary”](#) on page 192

WS-Federation Configuration Steps

- [“Selecting a Protocol”](#) on page 138
- [“General Information”](#) on page 139
- [“Setting an Assertion Lifetime”](#) on page 141
- [“Configuring Web SSO”](#) on page 143
- [“Configuring Credentials”](#) on page 184
- [“Connection Activation and Summary”](#) on page 192

SAML 1.x Configuration Steps

- [“Selecting a Protocol”](#) on page 138
- [“Importing Metadata”](#) on page 138
- [“General Information”](#) on page 139
- [“Setting an Assertion Lifetime”](#) on page 141
- [“Configuring Web SSO”](#) on page 143
- [“Configuring Credentials”](#) on page 184
- [“Connection Activation and Summary”](#) on page 192

For more information, see [“Navigating the Administrative Console”](#) on page 65.



Tip: You must completely configure a connection before you can save it on the Activation & Summary screen. Until then, the configuration is temporary and can be lost; the console times out after several minutes of inactivity. Before reaching Activation & Summary, however, you can click **Save Draft**, which is available on most screens after you enter General Information (see [“Console Buttons”](#) on page 67).

Selecting a Protocol

If your federation deployment supports multiple protocols, the first step in setting up a connection is to choose the applicable protocol.

If your partner's deployment supports multiple protocols and you will communicate using more than one, then you must set up a separate connection for each protocol.

SAML2.0 Configuring SP Connection	
Help Support About Logout (Administrator)	
Main SP Connection	
* Role & Protocol Import Metadata General Info Assertion Lifetime SAML Profiles Activation & Summary	
As an IdP, you are making a connection to a partner SP. For this connection, choose among the federation protocols you have enabled.	
Connection Type	SP
Protocol	SAML v2.0

- To continue, select the protocol for this connection, if needed, and click **Next**.

For information on enabling or disabling protocol support, see “[Choosing Roles and Protocols](#)” on page 74.

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](#)” on page 15.

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 XML 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 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 92).

SAML2.0 Configuring 'saml2' SP Connection
 [Help](#) | [Support](#) | [About](#) | [Logout \(Administrator\)](#)

[Main](#) | **SP Connection**

[Role & Protocol](#) | **[General Info](#)** | [Assertion Lifetime](#) | [SAML Profiles](#) | [Web SSO](#) | [Credentials](#) | [Activation & Summary](#)

This information identifies your partner's unique connection identifier (Connection ID). Optionally, you can specify a Virtual Server ID for *your own server* to use when communicating with this partner. If set, the virtual ID will be used in place of the unique protocol identifier configured for your server in Local Settings. The Base URL may be used to simplify configuration of partner endpoints.

Partner's Entity ID (Connection ID)	<input type="text" value="saml2"/> *
Virtual Server ID	<input type="text"/>
Base URL	<input type="text" value="https://pingidentity.com:9031"/>
Company	<input type="text"/>
Contact Name	<input type="text"/>
Contact Number	<input type="text"/>
Contact Email	<input type="text"/>
Logging Mode	<input type="radio"/> None <input checked="" type="radio"/> Standard <input type="radio"/> Enhanced <input type="radio"/> Full

Field Descriptions

Field	Definition
Partner's Entity ID/ Audience/ Partner's Realm (Connection ID)	(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 94).

Field	Definition
Virtual Server ID	<p>Enter a unique server ID in this field if you want to identify <i>your</i> server to this connection partner using an ID other than the one you specified under Server Settings (see “Specifying Federation Information” on page 75).</p> <p>For information about Virtual Server IDs, see “Federation Server Identification” on page 47.</p>
Base URL	The fully qualified hostname and port on which your partner’s federation deployment runs (e.g., <code>https://www.pingidentity.com:9031</code>). This entry is an optional convenience, allowing you to enter relative paths to specific endpoints, instead of full URLs, during the configuration process.
Company	The name of the partner company to which you are connecting.
Contact Name	The contact person at the partner company.
Contact Number	The phone number of the contact person at the partner company.
Contact Email	The email address for the contact person at the partner company.
Logging Mode	The level of transaction logging applicable for this connection (see “Runtime Transaction Logging” on page 92). Note that you can override connection logging mode settings globally from the connections list (see “Using the Connection List Screen” on page 134).

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** in the steps list.

For a new connection:

- Fill in the information needed and click **Next**.

Connection ID is required and must be the unique identifier of your connection partner (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 47).

For an existing connection:

- If you are editing existing information, modify the fields as needed and click **Save**.

Setting an Assertion Lifetime

The SAML and WS-Federation specifications 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 time stamp.

These settings are intended to compensate for unsynchronized IdP and SP system clocks.

Field Descriptions

Field	Definition
Minutes before	The amount of time before the assertion was issued during which it is to be considered valid.
Minutes after	The amount of time after the assertion was issued during which it is to be considered valid.

To change the default times:

- (Optional) Edit the desired setting(s) and click **Next** or **Save**.

Choosing SAML Profiles

A profile is the SAML message-interchange scenario that you and your federation partner have agreed to use (see “[Federation Planning Checklist](#)” on page 46). For SAML 2.0, PingFederate supports all IdP- and SP-initiated Web SSO and SLO profiles as well the Attribute Query profile (see “[Attribute Query and XASP](#)” on page 31).

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 “[SAML 1.x Profiles](#)” on page 16, “[SAML 2.0 Profiles](#)” on page 20, and “[Attribute Query and XASP](#)” on page 31.

You select the profiles you want to use with PingFederate from the SAML Profiles screen.



Note: For SAML 1.x, IdP-initiated SSO is assumed.

SAML 2.0 Configuring 'Partner2:entityId' SP Connection [Help](#) | [Support](#) | [About](#) | [Logout \(Administrator\)](#)

[Main](#) | [SP Connection](#)

✓ [Role & Protocol](#) | ✓ [Import Metadata](#) | ✓ [General Info](#) | ✓ [Assertion Lifetime](#) | ✗ [SAML Profiles](#)
| [Web SSO](#) | [Activation & Summary](#)

A SAML Profile defines what kind of messages may be exchanged between an Identity Provider and a Service Provider, and how the messages are transported ("bindings"). As an IdP, you configure this information for your SP connection.

Single Sign-On (SSO) Profiles	Single Logout (SLO) Profiles	Other Profiles
<input checked="" type="checkbox"/> IdP-Initiated SSO	<input checked="" type="checkbox"/> IdP-Initiated SLO	<input type="checkbox"/> Attribute Query
<input checked="" type="checkbox"/> SP-Initiated SSO	<input checked="" type="checkbox"/> SP-Initiated SLO	



Note: The SAML 1.x screen does not include choices for SLO—specifications do not support it. Also, the SP-Initiated SSO profile does not require any configuration for SP connections under SAML 1.x; your server supports this profile internally.

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 **SAML Profiles** in the steps list.

To begin configuring SAML profiles:

1. Select the profile(s) applicable to this connection and click **Next**.
2. To configure selected SSO/SLO profiles, click the **Configure Web SSO** button on the **Web SSO** screen.
See “[Configuring Web SSO](#)” on page 143.
3. To configure the Attribute Query profile, click the **Configure Attribute Query** button on the Attribute Query screen.

For SAML 2.0 connections that support SLO you must first select at least one SSO profile. You may then configure IdP- or SP-initiated SLO profiles or both, regardless of your SSO configuration.

The SAML 1.x specifications do not support SLO.

You can configure the profiles you need one at a time or all together. PingFederate will present you with the correct configuration steps to fit your

choices. Steps that apply to one profile often apply to others and are reused automatically across profiles.

Configuring Web SSO

The Web SSO screen displays the SSO profile choices you made in the previous step (see “[Choosing SAML Profiles](#)” on page 141).

SAML2.0 Configuring 'localhost:default:entityId' SP Connection [Help](#) | [Support](#) | [About](#) | [Logout \(Administrator\)](#)

[Main](#) | **SP Connection**

[Role & Protocol](#) |
 [General Info](#) |
 [Assertion Lifetime](#) |
 [SAML Profiles](#) |
 *** Web SSO** |
 [Credentials](#) |
 [Activation & Summary](#)

For each Web SSO profile selected, configure the necessary settings.

Single Sign-on (SSO) Profiles		Single Logout (SLO) Profiles	
IdP-Initiated SSO	Not Configured	IdP-Initiated SLO	Not Configured
SP-Initiated SSO	Not Configured	SP-Initiated SLO	Not Configured

[Configure Web SSO](#)

For profile configuration you will need to know:

- The type of subject name identifier you have agreed to send in assertions to this SP partner (see “[Identity Mapping](#)” on page 37).
- User attributes you and your IdP partner have agreed should be contained in the SAML assertion (see “[Attribute Contracts](#)” on page 41).
- Optionally, any additional attributes from data stores at your site that are needed to fulfill the attribute contract (see “[Attribute Contracts](#)” on page 41), as well as the location of the data stores and query parameters.
- Association of IdP adapter(s) with the connection including what user attributes are to be passed in from the user’s security session.
- The transport configurations ([bindings](#)) that you will use to send and receive data for SAML connections.
- Authentication and verification requirements for [inbound](#) and [outbound](#) SOAP back-channel transactions for profiles using the [artifact](#) binding, including HTTP Basic credentials and/or SSL certificate information. You can also choose to use digital signatures for back-channel messages, either in conjunction with one or more authentication methods or in place of authentication.



Note: As an IdP, the term “inbound” refers to the profile configuration for receiving SOAP messages such SLO requests and responses or artifact resolution requests. “Outbound” refers to the profile configuration for sending SOAP messages.

- For SSO profiles, the URL(s) of your SP's [Assertion Consumer 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\(s\)](#).
- 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.

The following sections provide more information on requirements for each SAML profile.



Note: For information about individual steps, see the list under “[SSO/SLO Profile Configuration Steps](#)” on page 146.

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 43).

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 signing assertions (not always required for the artifact binding)
- The URL(s) of your partner's [Assertion Consumer Service\(s\)](#).

To configure IdP-initiated SSO:

1. For SAML 2.0 connections select **IdP-Initiated SSO** on the SAML Profiles screen.

For SAML 1.x connections IdP-initiated SSO is assumed.

See “[Choosing SAML Profiles](#)” on page 141.

2. Follow the configuration steps.

See “[SSO/SLO Profile Configuration Steps](#)” on page 146 and “[Configuring Credentials](#)” on page 184.

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](#)” on page 20).

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 here; when the SP sends an authentication request to your SSO Service endpoint, PingFederate will handle the response automatically.

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, a combination of two of them, or use all of the mechanisms.

To configure SP-initiated SSO:

1. Select **SP-Initiated SSO** on the SAML Profiles screen.
See [“Choosing SAML Profiles”](#) on page 141.
2. Follow the configuration steps.
See [“SSO/SLO Profile Configuration Steps”](#) on page 146 and [“Configuring Credentials”](#) on page 184.

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.



Note: SLO is not supported by the SAML 1.x specifications.

For more information about SLO, see [“Single Logout”](#) on page 31.

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\(s\)](#)

To configure IdP-initiated SLO:

1. Select **IdP-Initiated SLO** on the SAML Profiles screen.
See [“Choosing SAML Profiles”](#) on page 141.

2. Follow the configuration steps.

See [“SSO/SLO Profile Configuration Steps”](#) on page 146 and [“Configuring Credentials”](#) on page 184.

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.



Note: SLO is not supported by the SAML 1.x specifications.

For more information about SLO, see [“Single Logout”](#) on page 31.

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\(s\)](#)
- To resolve artifacts from the SP, 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

To configure SP-initiated SLO:

1. Select **SP-Initiated SLO** on the SAML Profiles screen.

See [“Choosing SAML Profiles”](#) on page 141.

2. Follow the configuration steps.

See [“SSO/SLO Profile Configuration Steps”](#) on page 146 and [“Configuring Credentials”](#) on page 184.

SSO/SLO Profile Configuration Steps

The following sections provide information about profile configuration steps:

- [“Configuring Identity Mapping”](#) on page 147
- [“Creating an Attribute Contract”](#) on page 149
- [“IdP Adapter Mapping”](#) on page 151
- [“Specifying a Failsafe Attribute Source”](#) on page 168
- [“Mapping Default Attribute Contract Fulfillment”](#) on page 169
- [“Setting Assertion Consumer Service URLs \(SAML\)”](#) on page 171
- [“Setting a Default Target URL \(SAML 1.x\)”](#) on page 172
- [“Defining a Service URL \(WS-Federation\)”](#) on page 173

- [“Specifying SLO Service URLs \(SAML 2.0\)”](#) on page 174
- [“Choosing Allowable SAML Bindings \(SAML 2.0\)”](#) on page 175
- [“Configuring Signature Policy”](#) on page 178
- [“Configuring XML Encryption Policy \(SAML 2.0\)”](#) on page 178



Important: After modifying Web SSO profiles, you must click **Save** on the Web SSO screen.

Configuring Identity Mapping

PingFederate supports the option for 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 37). 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 coordinate in advance on which option to use (see [“Federation Planning Checklist”](#) on page 46).

Your choices of name identifier depend on which protocol you are using:

- For information about SAML 2.0 selections, see [“SAML Name ID Selections”](#) next.
- For information about WS-Federation selections, see [“WS-Federation Name ID Selections”](#) on page 149.

SAML Name ID Selections

The allowable name identifiers for SAML connections are described below. These choices affect how SPs make use of [account mapping](#) or [account linking](#).

SAML2.0
Configuring 'Partner1:entityId' SP Connection
[Help](#) | [Support](#) | [About](#) | [Logout \(Administrator\)](#)

Main
SP Connection
SP Web SSO

✱ Identity Mapping | ✓ Attribute Contract | ✓ IdP Adapter Mapping | ✓ Assertion Consumer Service URL | ✓ SLO Service URLs | ✓ Allowable SAML Bindings | ✓ Artifact Lifetime | ✓ Artifact Resolver Locations | ✓ Signature Policy | ✓ Encryption Policy | ✓ Summary

Identity mapping is the process in which users authenticated by the IdP are associated with user accounts local to the SP. Select the type of name identifier that you will send to the SP. Your selection may affect the way that the SP will look up and associate the user to a specific local account.

☒ **Standard:** Send the SP a known attribute value as the name identifier. The SP will often use account mapping to identify the user locally.

☐ **Pseudonym:** Send the SP a unique, opaque name identifier that preserves user privacy. The identifier cannot be traced back to the user's identity at this IdP and may be used by the SP to make a persistent association between the user and a specific local account. The SP will often use account linking to identify the user locally.

☐ Include attributes in addition to the pseudonym.

☐ **Transient:** Send the SP an opaque, temporary value as the name identifier.

☐ Include attributes in addition to the transient identifier.

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 40).



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) or opaque persistent name identifier (“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 128).

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 38).

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**.

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 **Web SSO** in the steps list.
3. Click the **Configure Web SSO** button on the Web SSO screen.
4. Click **Identity Mapping** in the steps list.

WS-Federation Name ID Selections

A name identifier is a uniquely identifying user attribute.

- ▶ Make one of the selections described below:
 - **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 Principle 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 169).

Creating an Attribute Contract

An attribute contract is the set of user attributes that you and your partner have agreed will be sent in a SAML assertion for this connection (see [“Attribute Contracts”](#) on page 41). You identify these attributes on this screen.

If you are sending a “standard” name identifier (see [“Configuring Identity Mapping”](#) on page 147), 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 165).



Note: Creating an attribute contract is optional if you are sending either a pseudonym or a transient identifier to your connection partner (see “Configuring Identity Mapping” on page 147).

The screenshot shows the 'Configuring 'Partner1:entityId' SP Connection' page in the SAML2.0 section. The breadcrumb trail is: Main > SP Connection > SP Web SSO. A list of configuration steps is shown: Identity Mapping (checked), Attribute Contract (selected with a star), IdP Adapter Mapping (checked), Assertion Consumer Service URL (checked), SLO Service URLs (checked), Allowable SAML Bindings (checked), Artifact Lifetime (checked), Artifact Resolver Locations (checked), Signature Policy (checked), Encryption Policy (checked), and Summary (checked). A green box explains: 'An Attribute Contract is a set of user attributes that this server will send in the assertion.' Below this is a table titled 'Attribute Contract' with one row: SAML_SUBJECT. Underneath is a section 'Extend the Contract' with a table listing attributes: email, fname, and lname. Each attribute has 'Edit / Delete' links. At the bottom is an 'Add' button next to a text input field.

Attribute Contract	
SAML_SUBJECT	

Extend the Contract	Action
email	Edit / Delete
fname	Edit / Delete
lname	Edit / Delete
<input type="text"/>	Add

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 **Web SSO** in the steps list.
3. Click the **Configure Web SSO** button on the Web SSO screen.
4. Click **Attribute Contract** in the steps list.

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 “Configuring Identity Mapping” on page 147).

To add an attribute:

- Enter the attribute name in the text box and click **Add**.
Attribute names are case-sensitive and must correspond to the names configured by your federation partner.

To modify an attribute name:

1. Click **Edit** under Action for the Attribute name.
2. Edit the name 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** for the Attribute Name.

IdP Adapter Mapping

IdP adapters are responsible for handling user authentication as part of an SSO operation (see “[Integration Kits and Adapters](#)” on page 39). Map one or more IdP adapter instances into each SP connection so that when a user authenticates with a particular external Identity Management System (IDM) 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 you 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 one or more instances of IdP adapters. If you have not yet configured an instance of the IdP adapter you intend to use within this SP connection, see “[Configuring IdP Adapters](#)” on page 124.

SAML2.0 Configuring 'Partner1:entityId' SP Connection [Help](#) | [Support](#) | [About](#) | [Logout \(Administrator\)](#)

[Main](#) | [SP Connection](#) | [SP Web SSO](#)

✓ Identity Mapping | ✓ Attribute Contract | ✗ **IdP Adapter Mapping** | ✓ Assertion Consumer Service URL | ✓ SLO Service URLs | ✓ Allowable SAML Bindings | ✓ Artifact Lifetime | ✓ Artifact Resolver Locations | ✓ Signature Policy | ✓ Encryption Policy | ✓ Summary

PingFederate uses IdP adapters to authenticate users to your partners. Users may be authenticated by one of several different adapters, so map an adapter instance for each IDM system on your server.

Adapter Instance Name	Action
<input type="button" value="Map New Adapter Instance..."/>	

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 **Web SSO** in the steps list.
3. Click the **Configure Web SSO** button on the Web SSO screen.
4. Click **IdP Adapter Mapping** in the steps list.

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** (see the next section).

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 124).

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 153).

SAML2.0 Configuring 'saml2' SP Connection [Help](#) | [Support](#) | [About](#) | [Logout \(Administrator\)](#)

[Main](#) | [SP Connection](#) | [SP Web SSO](#) | [IdP Adapter Mapping](#)

* [Adapter Instance](#) | [Assertion Mapping](#) | [Attribute Contract Fulfillment](#) | [Summary](#)

Select an IdP adapter instance that may be used to authenticate users for this partner. Attributes returned by the adapter instance you choose (the Adapter Contract) may be used to fulfill the Attribute Contract with your partner.

Adapter Instance: idpadapter ▼

Adapter Contract

email

userId

[Manage Adapter Instances...](#)

- 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:

1. Click a connection name on the Main Menu.
Click **Manage All SP**, if needed, to see a full list of connections.
2. Click **Web SSO** in the steps list.
3. Click the **Configure Web SSO** button on the Web SSO screen.
4. Click **IdP Adapter Mapping** in the steps list.

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 “[Configuring Identity Mapping](#)” on page 147).

- Click an Adapter Instance Name or **Map New Adapter Instance**.

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 [“Integration Kits and Adapters”](#) on page 39).

The values supplied by the adapter you have chosen are shown under Adapter Contract on the Assertion Mapping screen.

The screenshot shows the 'SAML2.0 Configuring 'saml2' SP Connection' interface. At the top, there are links for 'Help', 'Support', 'About', and 'Logout (Administrator)'. Below this is a navigation bar with tabs: 'Main', 'SP Connection', 'SP Web SSO', 'IdP Adapter Mapping', and a partially visible 'Summary' tab. The 'IdP Adapter Mapping' tab is active, and within it, the 'Assertion Mapping' step is selected, indicated by a star icon. Below the navigation bar, there is a progress indicator showing 'Adapter Instance' (checked), 'Assertion Mapping' (active), 'Attribute Contract Fulfillment' (checked), and 'Summary' (checked). A green informational box states: 'You can choose to fulfill the Attribute Contract with your partner using either the values provided by the "PF4 Standard Adapter v1.1" adapter. Or, you can use these values plus additional attributes retrieved from local data stores.' Below this, the 'Adapter Contract' section lists 'email' and 'userId'. At the bottom, there are two radio button options: 'Retrieve additional attributes from data stores to fulfill the Attribute Contract' (which is selected) and 'Use only the Adapter Contract values in the SAML assertion'.

To reach this screen:

- Click a connection name on the Main Menu.
Click **Manage All SP**, if needed, to see a full list of connections.
- Click **Web SSO** in the steps list.
- Click the **Configure Web SSO** button on the Web SSO screen.
- Click **IdP Adapter Mapping** in the steps list.
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 [“Configuring Identity Mapping”](#) on page 147).
- Click an Adapter Instance Name or **Map New Adapter Instance**.
- Click **Assertion Mapping** in the steps list.
 - ▶ 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 154).
 - ▶ 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 169).



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 149 and “[IdP Adapter Mapping](#)” on page 151). If the attribute contract requires more information, determine whether your local data stores can supply it. (You can also choose to use text constants for certain information—see “[Mapping Default Attribute Contract Fulfillment](#)” on page 169.)

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 149). 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.

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.

Description	Type	Action
Add Attribute Source...		

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 **Web SSO** in the steps list.
3. Click the **Configure Web SSO** button on the Web SSO screen.

4. Click **IdP Adapter Mapping** in the steps list.

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 [“Configuring Identity Mapping”](#) on page 147).

5. Click an Adapter Instance Name or **Map New Adapter Instance**.

6. Click **Attribute Sources & User Lookup** in the steps list.

If this step is not listed, then this connection is configured to use adapter values only (see [“Selecting Assertion Mapping”](#) on page 153).

To configure an attribute source:

- Click **Add Attribute Source** and complete the setup steps (see [“Attribute Source Setup”](#) on page 155). (Note it is not configured to look up attributes from a data store.)

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

If you need to add an attribute source, refer to the following sections for configuration guidance:

1. See [“Selecting a Data Store”](#) (next section).
2. If you use a JDBC data store, see:
 - a. [“Selecting a JDBC Database Table and Columns”](#) on page 157
 - b. [“Configuring a Database Filter \(WHERE Clause\)”](#) on page 159

If you use a LDAP data store, see:

- a. [“Configuring an LDAP Directory Search”](#) on page 161
- b. [“Configuring an LDAP Filter”](#) on page 163

If you use a Custom data store, see:

- a. [“Configuring Custom Source Filters”](#) on page 165
- b. [“Selecting Custom Source Fields”](#) on page 165

3. See [“Attribute Contract Fulfillment”](#) on page 165.

Selecting a Data Store

This screen allows you to choose a data store from a previously configured list (see [“Managing Data Stores”](#) on page 77). 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 149).

SAML2.0 Configuring 'Partner1:entityId' SP Connection [Help](#) | [Support](#) | [About](#) | [Logout \(Administrator\)](#)

[Main](#) | [SP Connection](#) | [SP Web SSO](#) | [IdP Adapter Mapping](#) | [Attribute Sources & User Lookup](#)

[* Data Store](#) | [Database Table and Columns](#) | [Database Filter](#) | [Attribute Contract Fulfillment](#) | [Summary](#)

This server uses local data stores to retrieve supplemental attributes to be sent in an assertion. Specify an Attribute Source name that will distinguish this user lookup for the selected data store.

Attribute Source Description: *

Active Data Store: *

Data Store Type:

Field Descriptions

Field	Definition
Attribute Source Description	A name that uniquely identifies the specific user look-up query. Since attribute sources can be reused within this same connection, the description helps distinguish between lookups to the same data store.
Active Data Store	The location of the source to be used for looking up additional attributes. Examples: jdbc:mysql://10.10.1.80:3306 users.mycompany.com:5200
Data Store Type	Indicates whether the data store is JDBC, LDAP, or Custom.

To reach this screen:

- Click a connection name on the Main Menu.
Click **Manage All SP**, if needed, to see a full list of connections.
- Click **Web SSO** in the steps list.
- Click the **Configure Web SSO** button on the Web SSO screen.
- Click **IdP Adapter Mapping** in the steps list.
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 “[Configuring Identity Mapping](#)” on page 147).
- Click an Adapter Instance Name or **Map New Adapter Instance**.

- Click **Attribute Sources & User Lookup** in the steps list.

If this step is not shown, you have elected not to look up attributes in data stores (see “[Selecting Assertion Mapping](#)” on page 153).

- 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 77).

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 149). Only one table may be used as a source of data for a JDBC lookup.

Field Descriptions

Field	Definition
Schema	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.

Field	Definition
Table	The name of the table contained in the database. Use the drop-down to change the table.
Columns to return from SELECT	Displays selected table columns. Select the columns that are associated with the desired attributes you would like to return from the JDBC query.

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 **Web SSO** in the steps list.
3. Click the **Configure Web SSO** button on the Web SSO screen.
4. Click **IdP Adapter Mapping** in the steps list.
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 [“Configuring Identity Mapping”](#) on page 147).
5. Click an Adapter Instance Name or **Map New Adapter Instance**.
6. Click **Attribute Sources & User Lookup** in the steps list.
If this step is not shown, you have elected not to look up attributes in data stores (see [“Selecting Assertion Mapping”](#) on page 153).
7. Click the attribute source Description link.
8. Click **Database Table and Columns** from the steps list.

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.



Tip: To determine what attributes to look up during a query, click the **Attribute Contract to Fulfill** link to see what information must be collected (see [“Creating an Attribute Contract”](#) on page 149). Then determine what information is coming in from the session lookup adapter (see [“Selecting Assertion Mapping”](#) on page 153). 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] [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 157).

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 124).

You can also apply additional search criteria from your own database, using any other columns from the targeted table.



Tip: Click **“View List of ALL 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.

SAML2.0 Configuring 'Partner1:entityId' SP Connection
 [Help](#) | [Support](#) | [About](#) | [Logout \(Administrator\)](#)

[Main](#) | [SP Connection](#) | [SP Web SSO](#) | [IdP Adapter Mapping](#) | [Attribute Sources & User Lookup](#)

[Data Store](#) | [Database Table and Columns](#) | [Database Filter](#) | [Attribute Contract Fulfillment](#) |

Summary

Please supply a WHERE clause to filter the data from your table.

Where

USERID=' \${username} '
 *

Adapter Values

\${lname}
\${fname}
\${username}
\${email}
\${sessionId}

[View List of ALL Columns from "users" table](#)

Field Descriptions

Field	Definition
Where	WHERE clause statements conditionally select data from a table. Enter the WHERE clause statement in the space provided. For example: WHERE email='clive@company.com'.

To reach this screen:

- Click a connection name on the Main Menu.
Click **Manage All SP**, if needed, to see a full list of connections.
- Click **Web SSO** in the steps list.
- Click the **Configure Web SSO** button on the Web SSO screen.
- Click **Attribute Sources & User Lookup** in the steps list.
If this step is not shown, you have elected not to look up attributes in data stores (see [“Selecting Assertion Mapping”](#) on page 153).
- Click the attribute source Description link.
- 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 165).

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.

SAML2.0 Configuring 'Partner1:entityId' SP Connection [Help](#) | [Support](#) | [About](#) | [Logout \(Administrator\)](#)

[Main](#) | [SP Connection](#) | [SP Web SSO](#) | [IdP Adapter Mapping](#) | [Attribute Sources & User Lookup](#)

✓ [Data Store](#) | ✗ **[LDAP Directory Search](#)** | [LDAP Filter](#) | [Attribute Contract Fulfillment](#) | [Summary](#)

Please configure your directory search. This information, along with the attributes supplied in the Adapter Contract, will be used to fulfill the Attribute Contract.

Base DN

Search Scope

Root Object Class	Attribute	Action
<input type="text" value="contact"/>	<input type="text" value="cn"/>	<input type="button" value="Add Attribute"/>

[Attribute Contract to Fulfill](#)

Field Descriptions

Field	Definition
Base DN	The base distinguished name of where the beginning of the tree structure search begins. Searches look for information at or below this node level.
Search Scope	Specifies the node depth of the query, which begins at the Base DN. (Subtree, One level, or Object).

Field	Definition
Root Object Class	The class containing the attributes you want.
Attributes to return from search	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.

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 **Web SSO** in the steps list.
3. Click the **Configure Web SSO** button on the Web SSO screen.
4. Click **IdP Adapter Mapping** in the steps list.
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 [“Configuring Identity Mapping”](#) on page 147).
5. Click an Adapter Instance Name or **Map New Adapter Instance**.
6. Click **Attribute Sources & User Lookup** in the steps list.
If this step is not shown, you have elected not to look up attributes in data stores (see [“Selecting Assertion Mapping”](#) on page 153).
7. Click the attribute source Description link.
8. 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 155.

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})
```

The left side variable is an attribute you selected earlier (see “[Configuring an LDAP Directory Search](#)” on page 161).

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 124).

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**” to view a list from which to copy and paste.

For general information about search filters, consult your LDAP documentation.

SAML2.0

Configuring 'Partner1:entityId' SP Connection

[Help](#) | [Support](#) | [About](#) | [Logout \(Administrator\)](#)

[Main](#)
[SP Connection](#)
[SP Web SSO](#)
[IdP Adapter Mapping](#)
[Attribute Sources & User Lookup](#)

[Data Store](#) | [LDAP Directory Search](#) | [LDAP Filter](#) | [Attribute Contract Fulfillment](#) | [Summary](#)

Please enter a Filter for extracting data from your directory.

Filter

uid=\${username} *

Values

\${email}

\${fname}

\${lname}

\${sessionId}

\${username}

[View List of Available LDAP Attributes](#)

Field Descriptions

Field	Definition
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.

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 **Web SSO** in the steps list.
3. Click the **Configure Web SSO** button on the Web SSO screen.
4. Click **IdP Adapter Mapping** in the steps list.
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 [“Configuring Identity Mapping”](#) on page 147).
5. Click an Adapter Instance Name or **Map New Adapter Instance**.
6. Click **Attribute Sources & User Lookup** in the steps list.
If this step is not shown, you have elected not to look up attributes in data stores (see [“Selecting Assertion Mapping”](#) on page 153).
7. Click the attribute source Description link.
8. Click **LDAP Filter** from the steps list.
If you have not yet defined an LDAP data store, see [“Selecting a Data Store”](#) on page 155.

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 82).

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.

SAML2.0 Configuring 'Partner1:entityId' SP Connection [Help](#) | [Support](#) | [About](#) | [Logout \(Administrator\)](#)

[Main](#) |
 [SP Connection](#) |
 [SP Web SSD](#) |
 [IdP Adapter Mapping](#) |
 [Attribute Sources & User Lookup](#)

[Data Store](#) |
 [Configure Custom Source Filters](#) |
 [Configure Custom Source Fields](#) |
 [Attribute Contract Fulfillment](#) |
 [Summary](#)

Please select those fields you would like to use when mapping attributes to the Adapter Contract.

- ☐ SSN
- ☐ phone-number
- ☐ work-phone-number
- ☐ cell-phone-number
- ☐ address
- ☐ zip
- ☐ favorite-pet

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 149). 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.

SAML2.0 Configuring 'saml2' SP Connection [Help](#) | [Support](#) | [About](#) | [Logout \(Administrator\)](#)

[Main](#) | [SP Connection](#) | [SP Web SSO](#) | [IdP Adapter Mapping](#)

[Adapter Instance](#) | [Assertion Mapping](#) | [Attribute Sources & User Lookup](#) | [Failsafe](#)
[Attribute Source](#) | [Attribute Contract Fulfillment](#) | [Summary](#)

Fulfill your Attribute Contract with values from the session lookup adapter or hard-coded text values. These values will be used if the user cannot be located in data stores. Negotiate with your partner to determine what default values to send.

Attribute Contract	Source	Value
email	<div>Text</div>	<div>idp email</div>
SAML_SUBJECT	<div>Adapter</div>	<div>userId</div>

Map each Target 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 you are using (see [“Integration Kits and Adapters”](#) on page 39).
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 or JDBC attributes you identified for this Attribute Source (see [“Configuring an LDAP Directory Search”](#) on page 161, [“Selecting a JDBC Database Table and Columns”](#) on page 157, or [“Configuring Custom Source Filters”](#) on page 165).
- 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.
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:

1. Click a connection name on the Main Menu.
Click **Manage All SP**, if needed, to see a full list of connections.
2. Click **Web SSO** in the steps list.
3. Click the **Configure Web SSO** button on the Web SSO screen.
4. Click **IdP Adapter Mapping** in the steps list.
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 [“Configuring Identity Mapping”](#) on page 147).
5. Click an Adapter Instance Name or **Map New Adapter Instance**.
6. Click **Attribute Sources & User Lookup** in the steps list.
If this step is not shown, you have elected not to look up attributes in data stores (see [“Selecting Assertion Mapping”](#) on page 153).
7. Click the attribute source Description link.
8. Click **Attribute Contract Fulfillment** from the steps list.
If you have not yet defined a data store, see [“Selecting a Data Store”](#) on page 155).

To map attributes:

1. Choose a Source for each Target attribute.
2. Choose (or enter) a Value for each Attribute.
See [“Map each Target attribute to fulfill the Attribute Contract from one of these Sources:”](#) on page 166. All values must be mapped.
3. Click **Next**.

Using the OGNL Edit Screen

An inline editor is available for OGNL expressions. The editor checks for errors in an expression and allows you to enter input values and test the resulting output.

- To reach the OGNL editor, click **Edit** under Actions for the expression.



Important: If you make changes to the expression and want to save them, click **Update** under Actions. To discard changes, click the **Cancel** link under Actions; click the **Cancel** button near the bottom of the screen only if you wish to discard all changes you have made in the current task.

To test an expression:

1. Enter an input value in the Source textbox 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.

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 Web SSO 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.

The screenshot shows a web interface for configuring a SAML2.0 connection. At the top, it says "SAML2.0 Configuring 'saml2' SP Connection" with links for Help, Support, About, and Logout (Administrator). Below this is a navigation bar with tabs: Main, SP Connection, SP Web SSO, and IdP Adapter Mapping. The IdP Adapter Mapping tab is active. Under this tab, there are several sub-tabs: Adapter Instance, Assertion Mapping, Attribute Sources & User Lookup, Failsafe Attribute Source (which is highlighted), Attribute Contract Fulfillment, and Summary. A green box contains the text: "If the user is not found in any of the previous data stores, you can either provide a hard-coded set of attributes or not allow the SSO transaction to complete." Below this, there are two radio button options: "Send user to SP using default list of attributes" (which is selected) and "Abort the SSO transaction".

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 **Web SSO** in the steps list.
3. Click the **Configure Web SSO** button on the Web SSO screen.
4. Click **IdP Adapter Mapping** in the steps list.
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 “Configuring Identity Mapping” on page 147).
5. Click an Adapter Instance Name or **Map New Adapter Instance**.
6. Click **Failsafe Attribute Source** in the steps list.
This step appears only if you are using data stores (see “Selecting Assertion Mapping” on page 153).

To specify whether to use a failsafe attribute source:

- Click the relevant radio button to either use 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 168) or if you choose not to set up attribute sources (see [“Selecting Assertion Mapping”](#) on page 153).

Map each Target 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 you are using (see [“Integration Kits and Adapters”](#) on page 39).

For example, you might choose the adapter attribute username to map to SAML_SUBJECT.

- Expression

Values are derived from an expression written in the Object-Graph Navigation Language (OGNL), which is based on the Java programming language. OGNL expressions are useful for evaluating attribute values and returning information based on those values. 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("net-worth"),
#result=#result.replace("$",""),
#result=#result.replace(",",""),
#result < 500000 ? "bronze" :
#result < 1000000 ? "silver" : "gold"
```

You reference OGNL variables using the # symbol. PingFederate provides predefined OGNL variables for IdP-adapter attributes plus any attributes retrieved from data stores.

For data-store attributes, use this syntax (for example):

```
#this.get("ds.amount")
```

For attributes that contain a space or a special character, use this syntax (for example):

```
#this.get("net-worth")
```

For more information, see [“Using the OGNL Edit Screen”](#) on page 167.

For more information about OGNL, see the OGNL Web site, www.ognl.org.



Note: The PingFederate runtime engine uses OGNL version 2.6.7.

- 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.

There are a variety of other 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.

Attribute Contract	Source	Value
email	Text	idp email
SAML_SUBJECT	Adapter	userid

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 **Web SSO** in the steps list.
3. Click the **Configure Web SSO** button on the Web SSO screen.
4. Click **IdP Adapter Mapping** in the steps list.

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 [“Configuring Identity Mapping”](#) on page 147).

5. Click an Adapter Instance Name or **Map New Adapter Instance**.
6. Click **Attribute Contract Fulfillment** in the steps list.

If you are using data stores for attribute mapping and this step does not appear, see [“Specifying a Failsafe Attribute Source”](#) on page 168.

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 **Next**.

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 SAML Profiles](#)” on page 141).

SAML2.0

Configuring 'Partner1:entityId' SP Connection

[Help](#) | [Support](#) | [About](#) | [Logout \(Administrator\)](#)

[Main](#) | [SP Connection](#) | [SP Web SSO](#)

[Identity Mapping](#) | [Attribute Contract](#) | [IdP Adapter Mapping](#) | [Assertion Consumer Service URL](#) | [SLO Service URLs](#) | [Allowable SAML Bindings](#) | [Artifact Lifetime](#) | [Artifact Resolver Locations](#) | [Signature Policy](#) | [Encryption Policy](#) | [Summary](#)

☐ As the IdP, you send SAML assertions to the SP's **Assertion Consumer Service**. Depending on the situation, the SP may request that the SAML assertion be sent to one of several URLs, via different bindings. Please provide all the possible assertion consumer URLs below and select one to be the default.

Default	Index	Binding	Endpoint URL	Action
default	0	Artifact	/sp/ACS.saml2	Edit / Delete
	1	POST	/sp/ACS.saml2	Edit / Delete
<input type="checkbox"/>	<input type="text"/>	- SELECT -	<input type="text"/>	<input type="button" value="Add"/>

Field Descriptions

Field	Definition
Default (SAML 2.0)	A check in this checkbox indicates that the URL configuration in that row will be used as a default. You must select a default, even if only one endpoint is identified.
Index (SAML 2.0)	Uniquely identifies multiple ACS endpoints.
Binding	The method of transmission: POST or Artifact.
Endpoint URL	A location to which the assertion is sent, according to partner requirements.

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 **Web SSO** in the steps list.
3. Click **Configure Web SSO**.
4. Click **Assertion Consumer Service URL(s)** in the steps list.

To define an Endpoint URL:

1. Select the Binding your partner specifies for the Endpoint.
2. (Optionally) Enter an Index value — 0 or 1, for example.
3. Enter the fully qualified Endpoint URL or just a relative path if you have defined a base URL (see [“General Information”](#) on page 139).

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.
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.

This URL is required regardless of whether you expect your local application(s) to specify the target, which ensures that the server functions correctly during SSO events.

SAML 1.1 Configuring 'http://success.com' SP Connection
 [Help](#) | [Support](#) | [About](#) | [Logout \(Administrator\)](#)

[Main](#) | [SP Connection](#) | [SP Web SSO](#)

[✓ Identity Mapping](#) | [✓ Attribute Contract](#) | [✓ IdP Adapter Mapping](#) | [✓ Assertion Consumer Service URL](#) | [✗ Default Target URL](#) | [✓ Artifact Lifetime](#) | [✓ Signature Policy](#) | [✓ Encryption Policy](#) | [✓ Summary](#)

Proper functioning of SAML v1.x SP connections requires that a default target URL be specified in the case that the IdP application does not include one in its SSO request. This default URL represents the destination on the SP where the user will be directed.

Default Target URL

Field Descriptions

Field	Definition
Default Target URL	The URL of the target SP resource.

To reach this screen:

- Click a connection name on the Main Menu.
Click **Manage All SP**, if needed, to see a full list of connections.
- Click **Web SSO** in the steps list.
- Click **Configure Web SSO**.
- Click **Default Target URL** in the steps list.

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.

WS-Fed Configuring 'wsf' SP Connection [Help](#) | [Support](#) | [About](#) | [Logout \(Administrator\)](#)

[Main](#) | **SP Connection** | SP Web SSO

✓ Identity Mapping | ✓ Attribute Contract | ✓ IdP Adapter Mapping | ✗ **Service URL** | ✓ Summary

As the IdP, you send SAML assertions and SLO cleanup messages to the SP. Specify here the URL where the SP is expecting to receive these messages.

Endpoint URL *

- Enter the fully qualified URL or just the relative path if you have defined a base URL (see “[General Information](#)” on page 139).

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 145 or “[Configuring SP-Initiated SLO](#)” on page 146).

SAML 2.0 Configuring 'Partner1:entityId' SP Connection [Help](#) | [Support](#) | [About](#) | [Logout \(Administrator\)](#)

[Main](#) | **SP Connection** | SP Web SSO

✓ Identity Mapping | ✓ Attribute Contract | ✓ IdP Adapter Mapping | ✓ Assertion Consumer Service URL | ✗ **SLO Service URLs** | ✓ Allowable SAML Bindings | ✓ Artifact Lifetime | ✓ Artifact Resolver Locations | ✓ Signature Policy | ✓ Encryption Policy | ✓ Summary

As the IdP, you may send SAML logout messages to the SP's **Single Logout Service**. Depending on the situation, the SP may request that messages be sent to one of several URLs, via different bindings. Please provide the endpoints that you would like to use.

Binding	Endpoint URL	Response URL	Action
Redirect	/sp/SLO.saml2		Edit / Delete
POST	/sp/SLO.saml2		Edit / Delete
- SELECT -	<input type="text"/>	<input type="text"/>	<input type="button" value="Add"/>

* *

Field Descriptions

Field	Definition
Binding	The method of transmission: POST, Artifact, Redirect, or SOAP.

Field	Definition
Endpoint URL	A location to which logout messages are sent, according to SP requirements.
Response URL	(Optional) A location on this IdP to which logout responses are sent.

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 **Web SSO** in the steps list.
3. Click **Configure Web SSO**.
4. Click **SLO Service** URLs in the steps list.

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 139).
3. Optionally, 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 (see [“Choosing SAML Profiles”](#) on page 141).

SAML2.0 Configuring 'Partner1:entityId' SP Connection [Help](#) | [Support](#) | [About](#) | [Logout \(Administrator\)](#)

[Main](#) | [SP Connection](#) | [SP Web SSO](#)

[✓ Identity Mapping](#) | [✓ Attribute Contract](#) | [✓ IdP Adapter Mapping](#) | [✓ Assertion Consumer Service URL](#) | [✓ SLO Service URLs](#) | [✗ Allowable SAML Bindings](#) | [✓ Artifact Lifetime](#) | [✓ Artifact Resolver Locations](#) | [✓ Signature Policy](#) | [✓ Encryption Policy](#) | [✓ Summary](#)

When the SP sends messages, what SAML bindings do you want to allow?

☒ Artifact
☒ POST
☒ Redirect

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 **Web SSO** in the steps list.
3. Click **Configure Web SSO**.
4. Click **Allowable SAML Bindings** in the steps list.

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.

SAML2.0 Configuring 'saml2' SP Connection [Help](#) | [Support](#) | [About](#) | [Logout \(Administrator\)](#)

[Main](#) | [SP Connection](#) | [SP Web SSO](#)

[✓ Identity Mapping](#) | [✓ Attribute Contract](#) | [✓ IdP Adapter Mapping](#) | [✓ Assertion Consumer Service URL](#) | [✓ SLO Service URLs](#) | [✓ Allowable SAML Bindings](#) | [✗ Artifact Lifetime](#) | [✓ Artifact Resolver Locations](#) | [✓ Signature Policy](#) | [✓ Encryption Policy](#) | [✓ Summary](#)

Artifacts are meant to be short-lived tokens representing an issued message. For how long should the recipient of the artifact be permitted to retrieve the corresponding message?

Artifact Lifetime second(s) *

The default value is 60 seconds. You can change this 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:

1. Click a connection name on the Main Menu.
Click **Manage All SP**, if needed, to see a full list of connections.

2. Click **Web SSO** in the steps list.
3. Click **Configure Web SSO**.
4. Click **Artifact Lifetime** in the steps list.

This step appears only if you have selected the Artifact binding for either a Web 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 you partner. The locations are also known collectively under SAML specifications as the [Artifact Resolution Service](#).

SAML2.0 Configuring 'Partner1:entityId' SP Connection [Help](#) | [Support](#) | [About](#) | [Logout \(Administrator\)](#)

Main | SP Connection | SP Web SSO

[✓ Identity Mapping](#) |
 [✓ Attribute Contract](#) |
 [✓ IdP Adapter Mapping](#) |
 [✓ Assertion Consumer Service URL](#) |
 [✓ SLO Service URLs](#) |
 [✓ Allowable SAML Bindings](#) |
 [✓ Artifact Lifetime](#) |
 [✖ Artifact Resolver Locations](#) |
 [✓ Signature Policy](#) |
 [✓ Encryption Policy](#) |
 [✓ Summary](#)

Please provide the remote party URLs that you will use to resolve/translate the artifact and get the actual request.

Index	URL	Action
0	/sp/ARS.ssaml2	Edit / Delete
<input type="text"/>	<input type="text"/>	<input type="button" value="Add Artifact Resolution Service"/>

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 **Web SSO** in the steps list.
3. Click **Configure Web SSO**.
4. Click **Artifact Resolver Locations** in the steps list.

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 Artifact Resolution Service**.

The URL must be fully qualified (defining protocol, host, and port) unless you have entered a base URL (see [“General Information”](#) on page 139).

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 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 optional settings for digital signatures. The choices you make on this screen depend on your partner agreement (see “[Digital Signing Policy Coordination](#)” on page 44).

Digital signing is required for SAML Response messages, including assertions, from your site via POST (or Redirect for SAML 2.0). Optionally, SSO authentication requests from the SP (SP-initiated SSO) may be signed under SAML 2.0 specifications. Check this option if your partner agreement includes this option. (The option appears only if you have enabled SP-initiated SSO using the POST or redirect bindings.)

Assertions inside your SAML Responses may also be separately signed. If your partner agreement includes this option, click the relevant checkbox on this screen. (This is the only choice on the SAML 1.x screen.)

SAML2.0 Configuring 'saml2' SP Connection [Help](#) | [Support](#) | [About](#) | [Logout \(Administrator\)](#)

[Main](#) | **SP Connection** | [SP Web SSO](#)

✓ Identity Mapping | ✓ Attribute Contract | ✓ IdP Adapter Mapping | ✓ Assertion Consumer Service URL | ✓ SLO Service URLs | ✓ Allowable SAML Bindings | ✱ **Signature Policy** | ✓ Encryption Policy | ✓ Summary

Additional guarantees of authenticity may be agreed upon between you and your partner. For SP-initiated SSO, you can choose to require signed authentication requests sent via the POST or redirect bindings. You can also choose to sign assertions sent to this partner, regardless of the binding used.

☐ Require AuthN requests to be signed when received via the POST or Redirect bindings

☐ Always sign the SAML Assertion

► Make your choice(s) per requirements 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 that profile).

Note that “The entire assertion” selection on the Encryption Policy screen includes the SAML_SUBJECT and all attributes.

SAML2.0 Configuring 'saml2' SP Connection [Help](#) | [Support](#) | [About](#) | [Logout \(Administrator\)](#)

[Main](#) | **SP Connection** | [SP Web SSO](#)

✓ Identity Mapping | ✓ Attribute Contract | ✓ IdP Adapter Mapping | ✓ Assertion Consumer Service URL | ✓ SLO Service URLs | ✓ Allowable SAML Bindings | ✓ Signature Policy | ✱ **Encryption Policy** | ✓ Summary

Additional guarantees of privacy may be used between you and your partner. Specify an encryption policy for the exchange of SAML messages.

☒ The entire assertion

☐ One or more attributes

☐ SAML_SUBJECT (Name Identifier)

☐ email

☐ None

☒ Encrypt the SAML Subject in SLO messages to the SP

☒ Require that the SAML Subject be encrypted in SLO messages from the SP

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 **Web SSO** in the steps list.
3. Click the **Configure Web SSO** button on the Web SSO screen.
4. Click **Encryption Policy** in the steps list.

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 require 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 Web SSO screen.

Editing and Saving Web SSO Configurations

On the Summary screen you can review or edit your Web SSO configuration.



Important: When you finish editing existing profiles, you must click **Done** on the Summary screen and then **Save** on the Web SSO screen. For a new connection, click **Done** and then click **Next** on the Web SSO screen. Save the entire connection on the Activation screen (see “[Connection Activation and Summary](#)” on page 192).

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 dependent or other changes, do so and continue by clicking **Done** until you reach the Web SSO screen.

3. Click **Save** on the Web 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 profile (see “[Choosing SAML Profiles](#)” on page 141). Attribute queries are not dependent on single sign-on but may be used independently or in conjunction with Web SSO to provide flexibility in how a user authenticates with SP applications (see “[Attribute Query and XASP](#)” on page 31).

SAML2.0 Configuring 'Partner1:entityId' SP Connection [Help](#) | [Support](#) | [About](#) | [Logout \(Administrator\)](#)

[Main](#) | [SP Connection](#)

✓ [Role & Protocol](#) | ✓ [General Info](#) | ✓ [Assertion Lifetime](#) | ✓ [SAML Profiles](#) | ✓ [Web SSO](#) | ✗ [Attribute Query](#) | ✓ [Credentials](#) | ✓ [Activation & Summary](#)

The Attribute Query Profile supports SPs in requesting user attributes. Click the button below to configure the necessary settings to support this profile.

[Configure Attribute Query Profile](#)

- To continue, click the **Configure Attribute Query Profile** button.

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.

SAML2.0 Configuring 'Partner1:entityId' SP Connection [Help](#) | [Support](#) | [About](#) | [Logout \(Administrator\)](#)

[Main](#) | [SP Connection](#) | [Attribute Query](#)

* [Retrievable Attributes](#) | [Data Store](#) | [Configure Custom Source Filters](#) | [Configure Custom Source Fields](#) | [Attribute Mapping Fulfillment](#) | [Security Policy](#) | [Summary](#)

Specify the list of attributes that may be returned to the SP in the response to an attribute request.

Retrievable Attributes	Action
address	Edit / Delete
favorite-pet	Edit / Delete
home-phone-number	Edit / Delete
work-phone-number	Edit / Delete
zip	Edit / Delete
<input type="text"/>	Add

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 151). Therefore, you must identify a data store that contains the attributes on your system.

SAML2.0 Configuring 'Partner1:entityId' SP Connection [Help](#) | [Support](#) | [About](#) | [Logout \(Administrator\)](#)

[Main](#) | [SP Connection](#) | [Attribute Query](#) | **[Data Store](#)**

✓ [Retrievable Attributes](#) | ✖ [Data Store](#) | ✓ [Summary](#)

This server uses local data stores to retrieve user attributes in response to an attribute request.

Active Data Store: *

Data Store Type: None

[Manage Data Stores...](#)

Configuring Data Store Lookup

The process of configuring PingFederate to look up attributes in a data store for attribute-query responses is the same as that used for SSO Attribute Sources and User Lookup. For detailed information, see the step-by-step procedures in the sections indicated below. Note that the screen text may differ slightly and only one data store may be configured to supply user attributes.

- If you use a JDBC data store, see:
 - [“Selecting a JDBC Database Table and Columns”](#) on page 157
 - [“Configuring a Database Filter \(WHERE Clause\)”](#) on page 159
- If you use a LDAP data store, see:
 - [“Configuring an LDAP Directory Search”](#) on page 161
 - [“Configuring an LDAP Filter”](#) on page 163
- If you use a Custom data store, see:
 - [“Configuring Custom Source Filters”](#) on page 165
 - [“Selecting Custom Source Fields”](#) on page 165

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.

SAML2.0 Configuring 'pingfederate3:default:entityId' SP Connection [Help](#) | [Support](#) | [About](#) | [Logout \(Administrator\)](#)

[Main](#) | [SP Connection](#) | [Attribute Query](#)

[Retrievable Attributes](#) |
 [Data Store](#) |
 [Configure Custom Source Filters](#) |
 [Configure Custom Source Fields](#) |
 [Attribute Mapping Fulfillment](#) |
 [Security Policy](#) |
 [Summary](#)

[SPXASPTASKLET_SPATTRIBUTESOURCEMAPPINGSTATE]

Attribute Mapping	Source	Value
address	Custom	address
cell	Custom	cell-phone-number
favorite-pet	Custom	favorite-pet
home-phone-number	Custom	phone-number
SSN	Custom	SSN
work-phone-number	Custom	work-phone-number
zip	Custom	zip

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.

- 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.

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 184).

SAML2.0 Configuring 'Partner1:entityId' SP Connection [Help](#) | [Support](#) | [About](#) | [Logout \(Administrator\)](#)

[Main](#) | [SP Connection](#) | [Attribute Query](#)

[✓ Retrievable Attributes](#) |
 [✓ Data Store](#) |
 [✓ Configure Custom Source Filters](#) |
 [✓ Configure Custom Source Fields](#) |
 [✓ Attribute Mapping Fulfillment](#) |
 ✱ [Security Policy](#) |
 [✓ Summary](#)

Specify the attribute requester profile's security policy with your partner.

☒ Sign the Response
☐ Sign the Assertion
☐ Encrypt the Assertion
☒ Require signed Attribute Query
☒ Require an encrypted Name Identifier

To configure attribute-query security policy for this partner:

- Check or uncheck the 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 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 SAML Web SSO or Attribute Query configuration may involve any or all of the following:

- [Configuring Back-Channel Authentication](#)
- [Configuring Digital Signature Settings](#)
- [Selecting Signature Verification Certificates](#)
- [Selecting an Encryption Certificate \(SAML\)](#)
- [Selecting a Decryption Key \(SAML\)](#)

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.

SAML2.0 Configuring 'Partner1:entityId' IdP Connection [Help](#) | [Support](#) | [About](#) | [Logout \(Administrator\)](#)

[Main](#) | **IdP Connection** | **Credentials**

✱ **Back-Channel Authentication** | ✓ Digital Signature Settings | ✓ Signature Verification Certificate
 | ✓ Select XML Encryption Certificate | ✓ Select XML Decryption Key | ✓ Summary

You selected one or more bindings that require additional security for communication with your partner. Please ensure that security settings are properly configured.

Send to your partner:

- SOAP SLO messages
- Artifact resolution requests
- Attribute Query requests

[Configure](#)

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 **Credentials** in the steps list.
3. Click the **Configure Credentials** button.
4. Click **Back-Channel Authentication** in the steps list.

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 bindings 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 the SOAP Authentication Type on the next screen:

- Basic — you will enter SOAP Basic credentials on a later screen.
- SSL Certificate — you will specify the certificate on a later screen.

This option is enabled only if you have specified an endpoint that uses SSL.

You and your partner might also have agreed to require a valid certificate chain; if so, select the checkbox for that option.

- Use Digital Signatures . . . — you will sign the message.

You will be asked to select a signing certificate on a later screen.

All three options may be combined or used separately.

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 115). 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. Click **Next**.
3. 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.

All three options may be combined or used separately.

Click the checkbox to Require SSL if the connection will use the secure protocol for basic or digital signature authentication.

4. Click **Next**.
5. If you chose Basic at Step 3, 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.

6. 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 CAs](#)” on page 111).
 - 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.

7. Click **Next**.
8. If you chose anchored SSL certificate verification at [Step 6](#), 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.

9. If you chose unanchored SSL certificate verification at [Step 6](#), select the certificate you will use to validate the SSL connection.

If you have not yet imported the certificate into PingFederate, click **Manage Certificates**.

10. Click **Next**.
11. 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 SAML Profiles](#)” on page 141) whenever [outbound POST](#) or [Redirect](#) bindings are used. The step also is required if you chose to sign the SAML assertion, SAML response, or artifact resolution messages (see “[Configuring Back-Channel Authentication](#)” on page 185).

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 **Credentials** in the steps list.
3. Click the **Configure Credentials** button.
4. Click **Configure Digital Signing Settings** in the steps list.
This step does not appear if your 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 118).
2. (Optional) If you have agreed to send your public key with the SAML message, click the checkbox to implement this requirement.

Selecting Signature Verification Certificates

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.

Depending on your agreement with this SP, SSO assertions, SAML 2.0 artifacts, or SOAP messages might also require signatures.

Whenever signatures are required, you must import your partner's public key certificate into the PingFederate store for signature verification.



Tip: To prevent any interruption of service due to an expired certificate, you can ask your partner for a new certificate in advance and use it in the Secondary certificate field. The PingFederate server will use the primary certificate until it expires and then try the secondary.

SAML2.0 Configuring 'saml2' SP Connection [Help](#) | [Support](#) | [About](#) | [Logout \(Administrator\)](#)

[Main](#) | **SP Connection** | [Credentials](#)

✓ Digital Signature Settings | ✖ **Signature Verification Certificate** | ✓ Summary

Incoming SAML messages may be signed by your partner. Please select which certificate(s) to use when verifying these digital signatures. When multiple certificates are chosen, each certificate is tried from the top of the list down until the signature is verified.

Primary: - SELECT - *

Secondary: - SELECT -

[Manage Certificates...](#)



Note: This screen differs significantly between SAML 2.0 and SAML 1.x—see procedures below.

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 **Credentials** in the steps list.
3. Click the **Configure Credentials** button.
4. Click **Signature Verification Certificate** in the steps list.
If this step does not appear, then your configuration does not require a verification certificate.

To specify a verification certificate for SAML 2.0:

1. Select the certificate from the drop-down list.
If you have not yet imported the certificate into PingFederate, click **Manage Certificates**.
2. Optionally, 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 178).

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 183).

SAML2.0 Configuring 'saml2' SP Connection [Help](#) | [Support](#) | [About](#) | [Logout \(Administrator\)](#)

[Main](#) | [SP Connection](#) | [Credentials](#)

✓ [Digital Signature Settings](#) | ✓ [Signature Verification Certificate](#) | ✖ **Select XML Encryption Certificate** | ✓ [Select XML Decryption Key](#) | ✓ [Summary](#)

Please select the partner certificate to use when encrypting message content as well as the preferred block encryption and key transport algorithms. Only RSA keys can be used for XML encryption.

Block Encryption Algorithm	Key Transport Algorithm
<input checked="" type="radio"/> AES-128	<input checked="" type="radio"/> RSA-v1.5
<input type="radio"/> AES-256 (help)	<input type="radio"/> RSA-OAEP
<input type="radio"/> Triple DES	

- SELECT -

[Manage Certificates...](#)

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 **Credentials** in the steps list.
3. Click the **Configure Credentials** button.
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 178).

To identify the encryption certificate:

1. (Optional) Change the default settings under Block Encryption Algorithm and/or Key Transport Algorithm.

Note that the use of stronger AES encryption is subject to export control restrictions. The standard JRE distribution does not support this encryption. To use the strongest AES encryption, when permissible, download and install the “JCE [Java Cryptography Extension] Unlimited Strength Jurisdiction Policy Files” from <http://java.sun.com/products/jce/javase.html#UnlimitedDownload>.

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/)” 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:

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** in the steps list.
3. Click the **Configure Credentials** button.

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 178).

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 118).



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).

Connection Activation and Summary

When you finish setting up a connection, you may choose to activate it immediately. No messages are actually sent or received until your partner's federation gateway is also established and a user actually initiates an SSO or SLO event.



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.

To change a Connection Status:

- Select either Active or Inactive and then click **Save**.



Important: Be sure to click **Save**. Otherwise, the status will not be changed.

To modify a connection:

1. Click the heading above the information you want to change.
2. Change the information on the step screen and click **Done** or **Save**.
3. Change any dependent information on other screens if needed.
PingFederate identifies dependencies for you.
4. When you return to the Activation & Summary screen or to a task summary screen, click **Save**.



Important: Be sure to click **Save**. Otherwise, the connection will not be reconfigured.

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 38).

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 151).

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.

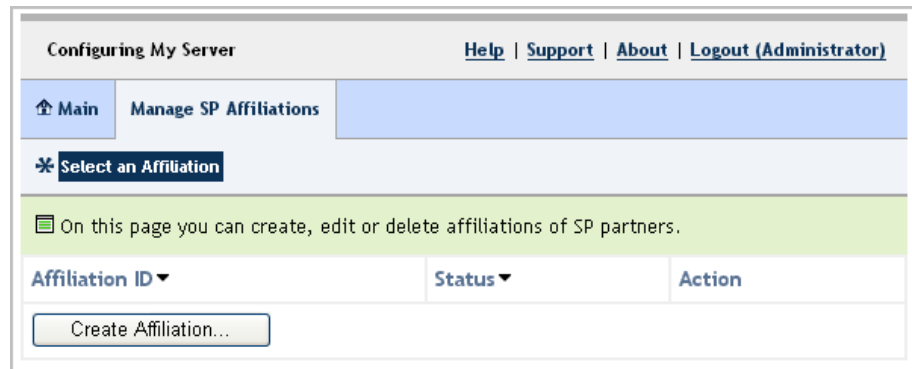
2. Click **Delete** under Action for the affiliation you want to delete.
3. Click **Save** to confirm the deletion (or click **undelete**).

To view or modify an affiliation:

- Click the affiliation name, or click **Manage All Affiliations** if the ID does not appear.

Using the Affiliations List Screen

You can manage SP affiliations from the **Select an Affiliation** screen.



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

You 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.

Field Descriptions

Field	Definition
Affiliation ID	A unique identifier for this affiliation.
Affiliation Owner	Any SAML 2.0 SP connection may serve as the Owner.

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.

The screenshot shows a web interface titled 'Configuring My Server'. At the top right are links for 'Help', 'Support', 'About', and 'Logout (Administrator)'. Below this is a navigation bar with 'Main', 'Manage SP Affiliations', and 'Create an Affiliation'. The 'Affiliation Membership' tab is selected, showing a progress bar with 'Import Metadata', 'Affiliation General Info', 'Affiliation Membership', and 'Activation & Summary'. A green box contains instructions: 'Manage the list of SP connections that are part of this affiliation. It is important that you configure each of the SP connections in an affiliation to generate opaque pseudonyms using the same adapter attributes.' Below this is a table with two columns: 'SP Connection ID' and 'Action'. The first row shows 'pingfederate3:default:entityId' with a 'Delete' link. The second row has a dropdown menu set to '- SELECT -' and an 'Add' button.

SP Connection ID	Action
pingfederate3:default:entityId	Delete
- SELECT -	Add

- To add an SP partner connection to the affiliation, select the ID from the drop-down list and click **Add**.



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 151).

- To remove a member of the affiliation, click **Delete** under Action for Connection ID and click **Save**.



Note: If you delete an affiliation member supplied by an imported metadata file and then save the affiliation, that connection ID will not appear in the drop-down list for re-adding in the future.

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**.

Service Provider Configuration

In an SP role, you manage connections to SAML or WS-Federate gateways located at 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 74).



Note: With PingFederate you can operate your enterprise in either an IdP, SP, or IdP Discovery role, or in any combination of roles (see [“Managing Server Settings”](#) on page 69).

Note that you configure only one connection per federation partner, even if you are integrating more than one Web application using the same federation protocol.

The integration of applications with PingFederate is a critical aspect of providing end-users with access to services across domains. This process is facilitated through the use of application integration kits and a robust Software Development Kit (see [“Integration Kits and Adapters”](#) on page 39).

This chapter contains the following topics:

- [“Application Integration Settings”](#) on page 198
- [“Federation Settings”](#) on page 208
- [“Configuring IdP Connections”](#) on page 211

Application Integration Settings

Under Application Integration Settings on the Main Menu, you configure SP Adapters that PingFederate uses to create user sessions that allow SSO access to your services. Optionally, you can also set Default URLs to which users may be directed during SSO or SLO, and you can look up system endpoints that IdP application developers may need to access PingFederate's SSO/SLO services.

Configuring SP Adapters

SP adapters are used to create a session for a user in order to provide SSO access to your application(s) or other resources (see [“Integration Kits and Adapters”](#) on page 39). You can configure multiple instances of adapters to accommodate different connection needs.



Note: If you are configuring an adapter for the first time as part of the post-installation process and using prepackaged adapters, see [“Standard Adapter Configuration”](#) on page 267 or [“LDAP Adapter Configuration”](#) on page 279.

If you configure more than one adapter instance, then you must map a target URL to at least one instance (see [“Mapping URLs to SP Adapter Instances”](#) on page 205).

SP adapter setup is available only if your server is configured as an SP. You reach the entry screen from the Main Menu.




Important: If you update an adapter JAR file after setting up connections using the same adapter, then you might need to reconfigure those connections. To find out, click each connection that uses the adapter (see [“Accessing Connections”](#) on page 211). Errors indicating reconfiguration points will be presented.

Configuring My Server
[Help](#) | [Support](#) | [About](#) | [Logout \(Administrator\)](#)

Main
Manage SP Adapter Instances

* Manage Adapter Instances

 PingFederate uses adapters to identify a user to your applications and/or identity management system based on attributes sent in an assertion. Create "instances" of adapters here to use within IdP connections for mapping these attributes to values needed by your local applications.

Adapter Instance Id	Adapter Instance Name	Adapter Type	Action
spadapter	spadapter	PF4 Standard Adapter v1.2	None Available - In Use

Create New Adapter Instance...

- ▶ You reach this screen by clicking **Adapters** under Application Integration Settings in My SP configuration.

To create a new adapter instance:

- ▶ Click **Create New Adapter Instance**.
See the next section.

To edit an adapter instance:

- ▶ Click the Adapter Instance Name link.

To delete an adapter instance:

1. Click **Delete** next to the Adapter 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 a connection.

2. Click **Save** to confirm the deletion.

Creating an Adapter Instance

On the Adapter Type screen, you begin creating an instance of an adapter that PingFederate will use for creating security sessions for your applications.

Configuring SP Adapter

[Help](#) | [Support](#) | [About](#) | [Logout \(Administrator\)](#)

[Main](#) | [Manage SP Adapter Instances](#) | [Create Adapter Instance](#)

[Adapter Type](#) | [SP Adapter Instance](#) | [Summary](#)

Please enter an Adapter Instance Name and Id, and select the Adapter Type.

Adapter Instance Name

Adapter Instance Id

Adapter Type

- SELECT -

[* Visit PingIdentity.com for additional adapter types](#)

Field Descriptions

Field	Definition
Adapter Instance Name	A descriptive name for the adapter instance—for example the target application or group of applications.
Adapter Instance ID	An internal identifier for the adapter instance. Must be alphanumeric with no spaces.
Adapter Type	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. In addition to bundled adapters (Standard and LDAP), a developer may deploy an adapter type before an administrator sets up a connection partner.

To reach this screen:

1. Click **Adapters** on the Main Menu.
2. Click **Create New Adapter Instance** on the Manage Adapter Instances screen.

To define an adapter instance:

1. Enter the Adapter Instance Name and Adapter Instance Id on the Adapter Type screen.
2. Select the Adapter Type from the drop-down menu.

If the adapter you need is not listed, click **Visit PingIdentity.com for Additional Adapters** to see if a suitable adapter is available from the PingFederate download site. You can also create your own adapter (see [“Integration Kits and Adapters”](#) on page 39).

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 [“Integration Kits and Adapters”](#) on page 39). For information about configuring the adapters packaged with PingFederate, see [“Standard Adapter Configuration”](#) on page 267 or [“LDAP Adapter Configuration”](#) on page 279.

4. Click **Done** on the Adapter Summary screen.
5. Click **Save** on the Manage Adapter Instances screen.

To view or modify adapter settings:

- Click the Adapter Instance Name.

To delete an adapter instance:

1. Click **Delete** next to the Adapter 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 differ vary according to the adapter you choose. These options are controlled by the adapter software (see [“Integration Kits and Adapters”](#) on page 39).

- For information about configuring the Standard Adapter, see [“Configuring the SP Standard Adapter”](#) on page 272.
- For information about configuring the LDAP Authentication Service, see [“Configuring the SP LDAP Adapter”](#) on page 284.

Configuring 'SPJava' SP Adapter
[Help](#) | [Support](#) | [About](#) | [Logout \(Administrator\)](#)

[Main](#)
[Manage SP Adapter Instances](#)
[Create Adapter Instance](#)

[Adapter Type](#) | [SP Adapter Instance](#) | [Adapter Actions](#) | [Extended Adapter Contract](#) | [Summary](#)

Complete the configuration necessary to set the appropriate security context for user sessions in your environment. This configuration was designed into the adapter for use at your site.

This is the standard adapter for PingFederate. The adapter uses a proprietary, secure token format (PFTOKEN) to transfer attributes between an application and the PingFederate server.

Field Name	Field Value	Description
Transfer method	<input type="radio"/> Cookie <input checked="" type="radio"/> Query parameter	How the PFTOKEN is transferred, either via a cookie or as a query parameter.
PFTOKEN holder name	SPJava	The name of the cookie, or the query parameter that contains the PFTOKEN. This name should be unique for each adapter instance.
Domain		The server domain, preceded by a period (e.g., .pingidentity.com). If no domain is specified, the value is obtained from the request
Cookie path	/	The path for the cookie that contains the PFTOKEN.
Password	****	The password used for generating a key to encrypt data.
Logout Service	http://localhost:8080/SpS	The URL to which the user is redirected for a Single Logout (SLO) event. This URL is part of an external application, which terminates the user session.
Authentication Service	http://localhost:8080/SpS	The URL to which the user is redirected for an SSO event. This URL is part of an external application, which performs user authentication.
Account Link Service		The URL to which the user is redirected for Account Linking. This URL is part of an external SP application, which performs user authentication and returns the local userid through PFTOKEN.

Show Advanced Fields

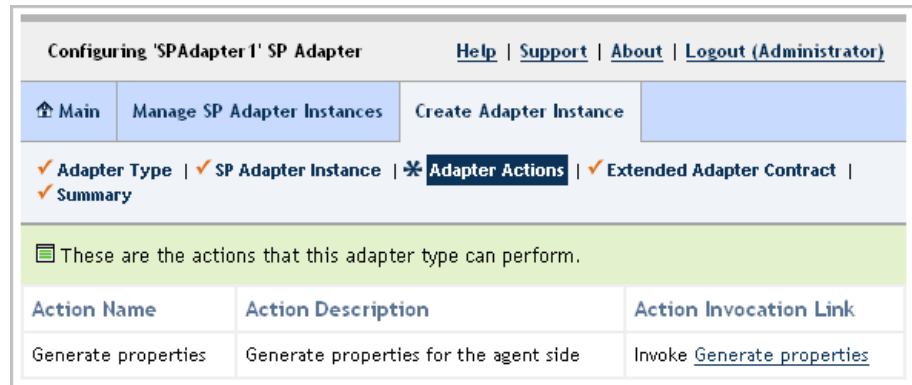
To reach this screen for editing:

1. On the Main Menu under Application Integration Settings for My SP Configuration, click **Adapters**.
2. Click an Adapter Instance Name.
3. Click **SP Adapter Instance**.

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 Standard Adapter, see [“Standard Adapter Configuration”](#) on page 267.
- For information about actions available using the LDAP Authentication Service, see [“LDAP Adapter Configuration”](#) on page 279.



To reach this screen for editing:

1. On the Main Menu under Application Integration Settings for My SP Configuration, click **Adapters**.
2. Click an Adapter Instance Name.
3. Click **Adapter Actions** (if available).

To generate a properties list:

- Click **Generate properties** 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.

Configuring 'SPAdapter1' SP Adapter
[Help](#) | [Support](#) | [About](#) | [Logout \(Administrator\)](#)

[Main](#) | [Manage SP Adapter Instances](#) | [Create Adapter Instance](#)

[Adapter Type](#) | [SP Adapter Instance](#) | [Adapter Actions](#) | [Extended Adapter Contract](#) | [Summary](#)

This adapter type supports the creation of an Extended Adapter Contract after initial deployment of the adapter instance. This Adapter Contract must be fulfilled using attributes from the SAML assertion combined with attributes returned from a data store lookup on this SP.

Adapter Contract	Action
userId	
Extend the Contract	
Email	Edit / Delete
FirstName	Edit / Delete
LastName	Edit / Delete
<input type="text"/>	Add

To reach this screen for editing:

1. On the Main Menu under Application Integration Settings for My SP Configuration, click **Adapters**.
2. Click an Adapter Instance Name.
3. Click **Extended Adapter 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 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 Adapter Instances screen.

To save an adapter instance:

1. Click **Done** on the Summary screen.
2. Click **Save** on the Manage 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 SP Adapter Instances](#)” in the next section).

Mapping URLs to SP 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 scenario that connects to multiple applications at your site (see “[Standards Support](#)” on page 13). 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 291). (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.

Configuring My Server

[Help](#) | [Support](#) | [About](#) | [Logout \(Administrator\)](#)

[Main](#)
Manage SP Adapter Instances

✓ Manage Adapter Instances | ✱ Map URLs to Adapter Instances

When you have more than one adapter instance defined, you must map a target URL at your site to at least one adapter instance. During SSO, the target URL requested will be compared against the URL(s) listed here until a match is found. Use a wildcard (*) to match multiple URLs to the same adapter instance.

URL	Adapter Instance	Action
<input type="text" value="http://www.target.com/*"/>	spadapter2	Update / Cancel
<input type="text" value=""/>	- SELECT -	Add Mapping

Field Descriptions

Field	Definition
URL	The target URLs that align with adapter instances that you have configured. The URLs instruct the PingFederate SP server to route session-creation processing to a particular application first. If the URL is not matched by the first adapter, the next adapter in the list is tried.
Adapter Instance (drop-down menu)	A selection of configured SP adapter instances.

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
- b. `http://yourapp.com/*` Adapter 2

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 198).

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 Name. 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 Name.
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; 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 will provide built-in destinations for the user. These Web pages are among the templates you can modify with your own branding or other information (see [“Using Velocity Templates”](#) on page 110).

The screenshot shows a web application titled "Configuring My Server". At the top right are links for "Help", "Support", "About", and "Logout (Administrator)". A navigation bar includes "Main" and "SP Default URLs". Below this is a section titled "SP Default URLs" with a green background. It contains instructions: "Enter values that affect the user's experience when executing SP-initiated Web SSO operations." and "Provide the default URL you would like to send the user to when Single Sign On (SSO) has succeeded." A text input field contains the URL "http://tyoes.corp.pingidentity.com:9090/SpSample". Below this, another instruction reads: "Provide the default URL you would like to send the user to when Single Logout (SLO) has succeeded."

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 may need to know the application endpoints to initiate transactions via PingFederate (see [“Integration Kits and Adapters”](#) on page 39).



Note: For specific parameters required or allowed for Application Endpoints, see [“Application Endpoints”](#) on page 289.

Federation Settings

Under Federation Settings on the Main Menu is a link you may need if your identity federation makes use of the SAML 2.0 XASP profile (see [“Attribute Query and XASP”](#) on page 31).

Also under Federation Settings, you can view protocol endpoints that your federation partners will 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”](#) on page 31). 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.

On the Attribute Requester Mapping screen, you map 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 74).

Configuring My Server
[Help](#) | [Support](#) | [About](#) | [Logout \(Administrator\)](#)

Main
Manage Attribute Requester Mapping

* Attribute Requester Mapping

During an Attribute Query request, the Issuer DN, if supplied in the request, is evaluated against the entries in the first table below in hierarchical order until a match is found. Use regular expressions to match different Issuer DNs to the same IdP Connection. If a match is found, the corresponding IdP connection is selected to issue the Attribute Query. If no Issuer DN match is found or if it is not provided, the Subject DN from the request is compared against the entries in the second table in a similar manner. If no Subject DN expression matches, then the default IdP connection is selected.

	Issuer DN	IdP Connection ID	Action
	ou=it,o=pingidentity,c=us	idp07	Edit / Delete
	<input type="text"/>	Partner1:entityId	Add

	Subject DN	IdP Connection ID	Action
▼	*,ou=resources,dc=corp,dc=pingidentity,dc=com	pingfederate3:default:entityId	Edit / Delete
▲	*,ou=employees,dc=corp,dc=pingidentity,dc=com	pingfederate3:default:entityId	Edit / Delete
	<input type="text"/>	Partner1:entityId	Add

Default
Partner1:entityId

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 regular expression may be used in any single entry. DN values must be entered in all lower-case characters.

- This screen is available from the Main Menu under Federation Settings in My SP Configuration.

To map attribute requesters to connection IDs:

- (Optional) Enter an Issuer DN when applicable, select a SAML 2.0 IdP Connection ID, and click **Add**.
Repeat this step as needed for additional DNs.
- Enter an Subject DN, select a SAML 2.0 IdP Connection ID, and click **Add**.
Repeat this step as needed for additional DNs.
- Select a Default IdP connection.

To edit a mapping:

- 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 and/or WS-Federation endpoints—a pop-up window displays only those endpoints related to the federation protocols you have chosen (see [“Choosing Roles and Protocols”](#) on page 74). These endpoints are built into PingFederate and cannot be changed.

Your federation partners need to know the applicable SP Services endpoints to communicate with your PingFederate server. Configured service endpoints are included in metadata export files (see [“Exporting Metadata”](#) on page 94).

The table below describes each endpoint:

Table 7: PingFederate SP Endpoints

Service	URL and Description
Single Logout Service (SAML 2.0)	/sp/SLO.saml2 The URL that receives and processes logout requests and responses.
Assertion Consumer Service (SAML 2.0)	/sp/ACS.saml2 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.
Artifact Resolution Service (SAML 2.0)	/sp/ARS.ssaml2 The SOAP endpoint that processes artifacts returned from a federation partner to retrieve the referenced XML message on the back channel.
Assertion Consumer Service (SAML 1.x)	/sp/acs.saml1 A SAML 1.x implementation URL that receives and processes assertions from an IdP.

Table 7: PingFederate SP Endpoints (Continued)

Service	URL and Description
Single Sign-on Service (WS-Federation)	<code>/sp/prp.wsrf</code> The WS-Federation implementation URL that receives and processes security tokens and SLO messages.

Configuring IdP Connections

As an SP, you configure connection settings to support the exchange of federation protocol messages (SAML or WS-Federation) with an IdP. These settings include:

- Attributes you expect to receive in an SSO assertion (if any) or 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 [“Standards Support”](#) on page 13.

To continue with the configuration, you and your federation partner must have decided this information in advance (see [“Federation Planning Checklist”](#) on page 46). Your federation partner must supply some connection settings and other information (see [“Configuration Data Exchange”](#) on page 49).

As an SP, you respond to user requests for SSO and SLO by creating or closing user sessions, respectively, via local applications. You integrate these applications with PingFederate by configuring them with SP [adapter](#) instances (see [“Configuring SP Adapters”](#) on page 198). 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 228). (No adapters are required for a connection that uses only the Attribute Query profile—see [“Configuring the Attribute Query Profile”](#) on page 253.)

If you intend to pass attribute values to an adapter 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 77).

Accessing Connections

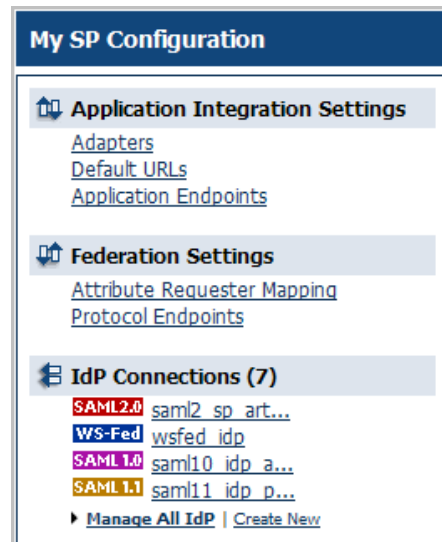
You can create or modify connections directly via the Main Menu. Note that the menu displays only four connections, listed under IdP Connections in order according to the most-recently modified. To view a list of all IdP connections, click the **Manage All IdP** link.

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 Connection List Screen” on page 213).



Note that long connection names are truncated for this display. The full connection names are displayed on the Select a Connection screen (see “Using the Connection List Screen” on page 213).

To begin configuring a new connection:

- ▶ Click **Create New** under IdP Connections on the Main Menu.
See “Configuration Steps” on page 214 for step-by-step information.



Tip: If you want to use a virtual ID for a second a connection to the same partner, the fastest way is to click **Manage All IdP** and copy the first connection (see “Using the Connection List Screen” on page 213). For information about virtual IDs, see “Federation Server Identification” on page 47.

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 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 Connection List Screen

From the Select a Connection screen, you can configure a new connection, modify or copy an existing connection or draft, or delete a connection (if it is not active). You can also globally override transaction logging levels set for individual connections or restore connection-based logging (see “[Runtime Transaction Logging](#)” on page 92).

The screenshot shows the 'Select a Connection' screen within the 'Manage IdP Connections' section. At the top, there are links for 'Help', 'Support', 'About', and 'Logout (Administrator)'. Below the navigation bar, a blue button labeled 'Select a Connection' is visible. A green informational box contains text about managing connections and filtering. Below this is a table with columns: Connection ID, Virtual ID, Protocol, Status, and Action. The table lists four connections: saml10, saml11, saml2, and wsfed. Each row shows the connection ID, a virtual ID (empty), a protocol (SAML 1.0, SAML 1.1, SAML 2.0, WS-Fed), a status (Active), and an action (In Use / Copy). Below the table is a 'Create Connection...' button. At the bottom, there is a 'Logging Mode Override' section with radio buttons for 'Off' (selected) and 'On'.

Connection ID	Virtual ID	Protocol	Status	Action
saml10		SAML 1.0	Active	In Use / Copy
saml11		SAML 1.1	Active	In Use / Copy
saml2		SAML 2.0	Active	In Use / Copy
wsfed		WS-Fed	Active	In Use / Copy

Create Connection...

Logging Mode Override
☒ Off
☐ On

To reach the Select a Connection screen:

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

To begin configuring a new connection:

- Click **Create Connection** on the Select a Connection screen.

See “[Configuring IdP Connections](#)” on page 211 for step-by-step information.



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 47.

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 217).
3. Make further changes needed for the new connection.

To edit a connection or continue working on a draft:

- Click the Connection ID link.

For a draft, you will return to where you left off.

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.

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 [“Standards Support”](#) on page 13). Also, for any given

connection, some configuration steps are required under the applicable protocol, while others are optional. The PingFederate administrative console determines the required steps and dynamically presents optional steps based on your selections.

The remainder of this section provides sequential information about every step you might encounter, 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 discussion always apply to the other protocols, unless otherwise indicated.

Use the lists and links (or page references) below to find specific information about steps that may apply to your connection requirements:

SAML 2.0 Web Configuration Steps

- [“Selecting a Protocol”](#) on page 216
- [“Importing Metadata”](#) on page 216
- [“General Information”](#) on page 217
- [“Choosing SAML Profiles”](#) on page 220
- [“Configuring Web SSO”](#) on page 221
- [“Configuring the Attribute Query Profile”](#) on page 253 (optional)



Note: You can configure the Attribute Query profile as a stand-alone connection or in conjunction with a SAML 2.0 Web SSO connection.

- [“Configuring Credentials”](#) on page 255
- [“Connection Activation and Summary”](#) on page 264

WS-Federation Configuration Steps

- [“Selecting a Protocol”](#) on page 216
- [“General Information”](#) on page 217
- [“Configuring Web SSO”](#) on page 221
- [“Configuring Credentials”](#) on page 255
- [“Connection Activation and Summary”](#) on page 264

SAML 1.x Configuration Steps

- [“Selecting a Protocol”](#) on page 216
- [“Importing Metadata”](#) on page 216
- [“General Information”](#) on page 217

- [“Choosing SAML Profiles”](#) on page 220
- [“Configuring Web SSO”](#) on page 221
- [“Configuring Credentials”](#) on page 255
- [“Connection Activation and Summary”](#) on page 264

For more information, see [“Navigating the Administrative Console”](#) on page 65.



Tip: You must completely configure a connection before you can save it on the Activation & Summary screen. Until then, the configuration is temporary and can be lost; the console times out after several minutes of inactivity. Before reaching Activation & Summary, however, you can click **Save Draft**, which is available on most screens after you enter General Information (see [“Console Buttons”](#) on page 67).

Selecting a Protocol

If your federation deployment supports multiple protocols, the first step in setting up a connection is to choose the applicable protocol.

If your partner’s deployment supports multiple protocols and you will communicate using more than one, then you must set up a separate connection for each protocol.

SAML2.0 Configuring 'Partner1:entityId' IdP Connection	
Help Support About Logout (Administrator)	
Main > IdP Connection	
* Role & Protocol General Info SAML Profiles Web SSO Attribute Query Credentials Activation & Summary	
<p>As an SP, you are making a connection to a partner IdP. For this connection, choose between the federation protocols you have enabled.</p>	
Connection Type	IdP
Protocol	SAML v2.0

- To continue configuring a new connection, select the applicable protocol (if needed) and click **Next**.

For information on enabling or disabling protocol support, see [“Choosing Roles and Protocols”](#) on page 74.

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”](#) on page 15.

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 XML 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 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 92).

SAML2.0 Configuring 'saml2' IdP Connection
 [Help](#) | [Support](#) | [About](#) | [Logout \(Administrator\)](#)

[Main](#) | **IdP Connection**

[Role & Protocol](#) | **General Info** | [SAML Profiles](#) | [Web SSO](#) | [Credentials](#) | [Activation & Summary](#)

This information identifies your partner's unique connection identifier (Connection ID). Optionally, you can specify a Virtual Server ID for *your own server* to use when communicating with this partner. If set, the virtual ID will be used in place of the unique protocol identifier configured for your server in Local Settings. The Base URL may be used to simplify configuration of partner endpoints.

Partner's Entity ID (Connection ID)	<input type="text" value="saml2"/> *
Virtual Server ID	<input type="text"/>
Base URL	<input type="text" value="https://my_hostname:9031"/>
Company	<input type="text"/>
Contact Name	<input type="text"/>
Contact Number	<input type="text"/>
Contact Email	<input type="text"/>
Error Message:	<div> An error has occurred in your transaction. Please contact your system administrator. </div>
Logging Mode	<input type="radio"/> None <input checked="" type="radio"/> Standard <input type="radio"/> Enhanced <input type="radio"/> Full

Field Descriptions

Field	Definition
Partner's Entity ID/ Issuer/ Partner's Realm (Connection ID)	(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 94).

Field	Definition
Virtual Server ID	Enter a unique server ID in this field if you want to identify <i>your</i> server to this connection partner using an ID other than the one you specified under Server Settings (see “Specifying Federation Information” on page 75). For information about Virtual Server IDs, see “Federation Server Identification” on page 47.
Base URL	The fully qualified hostname and port on which your partner’s federation deployment runs (e.g., https://www.pingidentity.com:9031). This entry is an optional convenience, allowing you to enter relative paths to specific endpoints, instead of full URLs, during the configuration process.
Company	The name of the partner company to which you are connecting.
Contact Name	The contact person at the partner company.
Contact Number	The phone number of the contact person at the partner company.
Contact Email	The email address for the contact person at the partner company.
Error Message	When an error occurs during a Web SSO operation on this server, the end user’s browser is redirected to an error page hosted within . The text you enter here is shown on that page and is intended to help the user understand what he/she should do next.
Logging Mode	The level of transaction logging applicable for this connection (see “Runtime Transaction Logging” on page 92). Note that you can override connection logging mode settings globally from the connections list (see “Using the Connection List Screen” on page 213).

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 **General Info** in the steps list.

For a new connection:

- Fill in the information needed and click **Next**.

Connection ID is required and must be the unique identifier of your connection partner (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 47).

For an existing connection:

- If you are editing existing information, modify the fields as needed and click **Save**.

Choosing SAML Profiles

A profile is the SAML message-interchange scenario that you and your federation partner have agreed to use (see “[Federation Planning Checklist](#)” on page 46). For SAML 2.0, PingFederate supports all IdP- and SP-initiated Web SSO and SLO profiles as well the Attribute Query profile (see “[Attribute Query and XASP](#)” on page 31).

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 “[SAML 1.x Profiles](#)” on page 16, “[SAML 2.0 Profiles](#)” on page 20, and “[Attribute Query and XASP](#)” on page 31.

You select SAML profiles in PingFederate from the SAML Profiles screen.



Note: For SAML 1.x, IdP-initiated SSO is assumed; the choice on this screen is limited to SP-initiated SSO.

Single Sign-On (SSO) Profiles	Single Logout (SLO) Profiles	Other Profiles
<input checked="" type="checkbox"/> IdP-Initiated SSO	<input checked="" type="checkbox"/> IdP-Initiated SLO	<input checked="" type="checkbox"/> Attribute Query
<input checked="" type="checkbox"/> SP-Initiated SSO	<input checked="" type="checkbox"/> SP-Initiated SLO	

For SAML 2.0 connections that support SLO you must first select at least one SSO profile. You may then configure IdP- or SP-initiated SLO profiles or both, regardless of your SSO configuration.

The SAML 1.x specifications do not support SLO.

You can configure the profiles you need one at a time or all together. PingFederate will present you with the correct configuration steps to fit your choices. Steps that apply to one profile often apply to others and are reused automatically across profiles.

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 **SAML Profiles** in the steps list.

To begin configuring SAML profiles:

1. Select the profile(s) applicable to this connection and click **Next**.
2. To configure selected SSO/SLO profiles, click the **Configure Web SSO** button on the **Web SSO** screen.
See “[Configuring Web SSO](#)” on page 221.
3. To configure the Attribute Query profile, click the **Configure Attribute Query** button on the Attribute Query screen.
See “[Configuring the Attribute Query Profile](#)” on page 253.

Configuring Web SSO

The Web SSO screen displays the SSO profile choices you made in the previous configuration step (see “[Choosing SAML Profiles](#)” on page 220).

SAML2.0 Configuring
'pingfederate3:default:entityId' IdP Connection
[Help](#) | [Support](#) | [About](#) | [Logout \(Administrator\)](#)

[Main](#) | **IdP Connection**

✓ [Role & Protocol](#) | ✓ [General Info](#) | ✓ [SAML Profiles](#) | ✗ **Web SSO** | ✓ [Attribute Authority](#) |
✓ [Credentials](#) | ✓ [Activation & Summary](#)

For each Web SSO Profile selected, configure the necessary settings.

Single Sign-on (SSO) Profiles		Single Logout (SLO) Profiles	
IdP-Initiated SSO	Configured	IdP-Initiated SLO	Configured
SP-Initiated SSO	Configured	SP-Initiated SLO	Configured

[Configure Web SSO](#)

For profile configuration you will need to know:

- The form of identity mapping you will use to provide access to your applications or resources (see “[Identity Mapping](#)” on page 37).
- User attributes you and your IdP partner have agreed should be contained in the SAML assertion (see “[Attribute Contracts](#)” on page 41).
- Optionally, any additional attributes from data stores at your site that are needed by the SP adapter you are using (see “[Adapter Contracts](#)” on page 41), as well as the location of the data stores and query parameters.

- Association of SP adapter(s) with the connection including what user attributes are passed to the user's session when it is created in the local application(s).
- The transport configurations ([bindings](#)) that you will use to send and receive data for SAML connections.
- Authentication and verification requirements for [inbound](#) and [outbound](#) SOAP back-channel transactions for profiles using the [artifact](#) binding, including HTTP Basic credentials and/or SSL certificate information. You can also choose to use digital signatures for back-channel messages, either in conjunction with one or more authentication methods or in place of authentication.



Note: As an SP, the term “inbound” refers to the profile configuration for receiving assertions, SOAP messages, SLO requests and responses, or artifacts. “Outbound” refers to the profile configuration for sending authentication requests, SOAP messages, SLO requests and responses, or artifacts.

- 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\(s\)](#).
- 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.

The following sections provide more information on requirements for each SAML profile.



Note: For information about individual steps, see the list under “[SSO/SLO Profile Configuration Steps](#)” on page 224.

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 43).

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, a combination of two of them, or use all of the 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 (see [“Single Sign-on”](#) on page 20).

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.

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, a combination of two of them, or use all of the 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.



Note: SLO is not supported by the SAML 1.x specifications.

For more information about SLO, see [“Single Logout”](#) on page 31.

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.



Note: SLO is not supported by the SAML 1.x specifications.

For more information about SLO, see [“Single Logout”](#) on page 31.

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

SSO/SLO Profile Configuration Steps

The following sections provide information about Web SSO and SLO profile configuration steps:

- [“Selecting Identity Mapping”](#) on page 225
- [“Creating an Attribute Contract”](#) on page 226
- [“Configuring Adapter Mapping and User Lookup”](#) on page 228
 - [“Selecting an Adapter Instance”](#) on page 229
 - [“Selecting an Adapter Data Store”](#) on page 230
 - [“Data Store Setup”](#) on page 231
 - [“Selecting a Data Store”](#) on page 232
 - [“Configuring Adapter Contract Fulfillment”](#) on page 241
- [“Specifying SSO Service URLs \(SAML\)”](#) on page 244
- [“Specifying a Service URL \(WS-Federation\)”](#) on page 246
- [“Specifying SLO Service URLs \(SAML 2.0\)”](#) on page 246
- [“Choosing Allowable SAML Bindings \(SAML\)”](#) on page 248
- [“Setting an Artifact Lifetime \(SAML 2.0\)”](#) on page 248
- [“Specifying Artifact Resolver Locations”](#) on page 249
- [“Configuring Signature Policy”](#) on page 250
- [“Configuring XML Encryption Policy \(SAML 2.0\)”](#) on page 251



Important: After modifying Web SSO profiles, you must click **Save** on the Web SSO screen.

Selecting Identity Mapping

PingFederate supports the option for 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 37). At the Identity Mapping step, you choose which method to use for a particular IdP connection. You and your partner may want to coordinate in advance on which option to use (see “[Federation Planning Checklist](#)” on page 46).

SAML2.0 Configuring 'Partner1:entityId' IdP Connection [Help](#) | [Support](#) | [About](#) | [Logout \(Administrator\)](#)

[Main](#) | **IdP Connection** | IdP Web SSO

* **Identity Mapping** | ✓ Attribute Contract | ✓ Adapter Mapping & User Lookup | ✓ SSO Service URLs | ✓ SLO Service URLs | ✓ Allowable SAML Bindings | ✓ Artifact Resolver Locations | ✓ Signature Policy | ✓ Encryption Policy | ✓ Summary

Identity mapping is the process whereby users authenticated by the IdP are associated with user accounts local to the SP. PingFederate supplies two modes for identity mapping of disparate user accounts between different domains. Choose which of these two styles to use to associate the user with a specific local account.

☒ **Account Mapping:** The IdP is sending a set of attributes that may be used to dynamically map the user to a specific local account.

☐ **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.

☐ The assertion includes attributes in addition to the unique name identifier.

If you are 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 40).



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) or opaque persistent name identifier (“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.

- ▶ 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:

1. Click a connection name on the Main Menu.
Click **Manage All IdP**, if needed, to see a full list of connections.
2. Click **Web SSO** in the steps list.
3. Click the **Configure Web SSO** button on the Web SSO screen.

Click Identity Mapping in the steps list.

Creating an Attribute Contract

An attribute contract is the set of user attributes that you and your partner have agreed will be sent in a SAML assertion for this connection (see “[Attribute Contracts](#)” on page 41). 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 Identity Mapping](#)” on page 225).

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 42).

SAML2.0 Configuring 'Partner3:entityId' IdP Connection [Help](#) | [Support](#) | [About](#) | [Logout \(Administrator\)](#)

[Main](#) | [IdP Connection](#) | [IdP Web SSO](#)

[Identity Mapping](#) | [Attribute Contract](#) | [Adapter Mapping & User Lookup](#) | [SSO Service URLs](#) | [SLO Service URLs](#) | [Allowable SAML Bindings](#) | [Signature Policy](#) | [Encryption Policy](#) | [Summary](#)

An Attribute Contract is a set of user attributes that the IdP will send in the assertion.

Attribute Contract		
SAML_SUBJECT		
Extend the Contract	Mask Values in Log	Action
email	<input type="checkbox"/>	Edit / Delete
<input type="text"/>	<input type="checkbox"/>	Add

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 **Web SSO** in the steps list.
3. Click the **Configure Web SSO** button on the Web SSO screen.
4. Click **Attribute Contract** in the steps list.

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 Identity Mapping](#)” on page 225).

To add an attribute:

1. Enter the attribute name in the text box.
Attribute names are case-sensitive and must correspond to the names configured by your federation partner.
2. Optionally, select the checkbox under Mask Values in Log.
3. Click **Add**.

To modify an attribute name or masking status:

1. Click **Edit** under Action for the Attribute name.
2. Edit the name and/or change the masking status, and then 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** for the Attribute Name.

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 local user to your applications based on attributes sent in an assertion. The server uses this information to create a local user context—in relation to your enterprise identity management (IDM) system, for example—that enables access to user-requested resources at your site (the “target”). (See [“Integration Kits and Adapters”](#) on page 39.)

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 map at least one adapter instance, which represents a specific deployment of an adapter in PingFederate. 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 SP Adapter Instances”](#) on page 205).

SAML2.0 Configuring 'Partner1:entityId' IdP Connection [Help](#) | [Support](#) | [About](#) | [Logout \(Administrator\)](#)

[Main](#) | [IdP Connection](#) | [IdP Web SSO](#)

✓ Identity Mapping | ✓ Attribute Contract | ✖ **Adapter Mapping & User Lookup** | ✓ SSO Service URLs | ✓ SLO Service URLs | ✓ Allowable SAML Bindings | ✓ Artifact Resolver Locations | ✓ Signature Policy | ✓ Encryption Policy | ✓ Summary

PingFederate uses session-creation adapters to identify a user to your applications and/or identity management system based on attributes sent in an assertion. Map an adapter instance for each target application on your system.

Adapter Instance Name	Action
Demo SP Adapter	Delete

[Map New Adapter Instance...](#)

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 **Web SSO** in the steps list.
3. Click the **Configure Web SSO** button on the Web SSO screen.
4. Click **Adapter Mapping & User Lookup** in the steps list.

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 within PingFederate.

The screenshot shows a web interface for configuring a SAML2.0 IdP connection. At the top, it says 'SAML2.0 Configuring 'saml2' IdP Connection' with links for Help, Support, About, and Logout (Administrator). Below this is a navigation bar with tabs: Main, IdP Connection, IdP Web SSO, and Adapter Mapping & User Lookup. The 'Adapter Instance' tab is selected, showing a sub-navigation bar with links: Adapter Instance, Adapter Data Store, Adapter Contract Fulfillment, and Summary. The main content area has a green box with the instruction: 'Select the adapter instance you would like to activate for incoming SAML messages from this partner.' Below this is a form with a label 'Adapter Instance' and a dropdown menu currently showing '- SELECT -'. Underneath is a section for 'Adapter Contract' with a button labeled 'Manage Adapter Instances...'.

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 198).

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 77).

SAML2.0 Configuring 'saml2' IdP Connection [Help](#) | [Support](#) | [About](#) | [Logout \(Administrator\)](#)

[Main](#) | [IdP Connection](#) | [IdP Web SSO](#) | [Adapter Mapping & User Lookup](#)

✓ [Adapter Instance](#) | ✱ [Adapter Data Store](#) | ✓ [Data Store](#) | ✓ [Summary](#)

You can fulfill the Adapter Contract by using only the attributes from the SAML assertion or by using these attributes to look up additional information from a local data store.

Attribute Contract

SAML_SUBJECT

email

☒ Use the SSO Assertion to lookup additional information

☐ Use only the attributes available in the SSO Assertion

- 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](#)”).

Or:

If you choose not to look up additional values, click the applicable button 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 241).



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 226 and “[Selecting an Adapter Instance](#)” on page 229). If the adapter requires more information, determine whether your local data stores can supply it. (You can also choose to use text constants for certain information—see “[Configuring Adapter Contract Fulfillment](#)” on page 241.)

Data Store Setup

If you need to look up additional values in a data store, follow the procedures described in these sections of the manual:

1. See “[Selecting a Data Store](#)” on page 232.
2. If you are using a JDBC data store, see:
 - a. “[Selecting a Database Table and Columns](#)” on page 233
 - b. “[Configuring a Database Filter \(WHERE Clause\)](#)” on page 234

If you are using a LDAP data store, see:

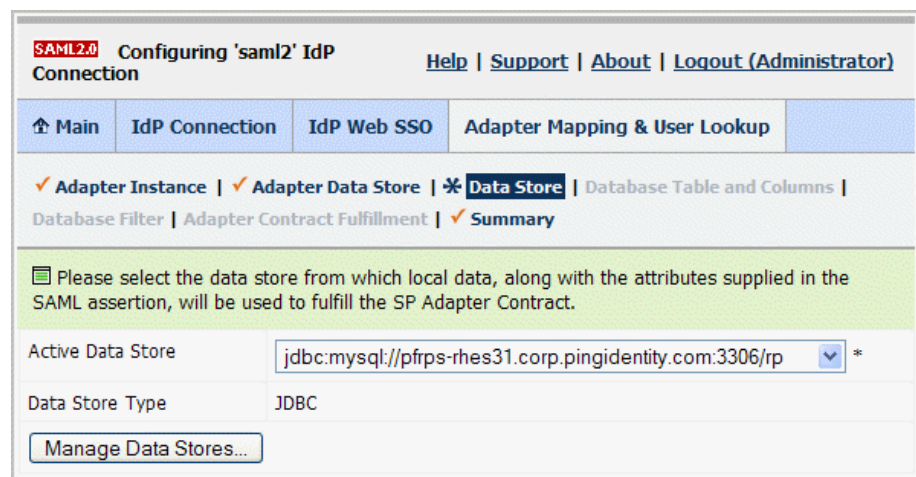
- a. “[Configuring an LDAP Directory Search](#)” on page 236
- b. “[Configuring an LDAP Filter](#)” on page 238

If you are using a Custom data store, see:

- a. [“Configuring Custom Source Filters”](#) on page 240
 - b. [“Selecting Custom Source Fields”](#) on page 240
3. See [“Configuring Adapter Contract Fulfillment”](#) on page 241

Selecting a Data Store

This screen allows you to choose a data store from a previously configured list (see [“Managing Data Stores”](#) on page 77). 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 228).



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 **Web SSO** in the steps list.
3. Click the **Configure Web SSO** button on the Web SSO screen.
4. Click **Adapter Mapping & User Lookup** in the steps list.
5. Click the adapter name.
6. Click **Data Store** in the steps list.

If this step is not present, then you have not chosen to use a data store to supplement assertion attributes (see [“Selecting an Adapter Data Store”](#) on page 230).

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 77).

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 are specifying 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 234).

Field Descriptions

Field	Definition
Schema	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.
Table	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.
Add Attribute	Adds the column to be executed in the SQL query to the data store to retrieve the attribute value.
(Drop-down menu)	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.

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 **Web SSO** in the steps list.
3. Click the **Configure Web SSO** button on the Web SSO screen.
4. Click **Adapter Mapping & User Lookup** in the steps list.
5. Click the adapter name.
6. Click **Database Tables and Columns** in the steps list.

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 232.

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.



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 229). Then determine what information is coming in from the assertion (see [“Creating an Attribute Contract”](#) on page 226). 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 149).

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 233).

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 ALL 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` 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.

Field Descriptions

Field	Definition
Where	WHERE clause statements conditionally select data from a table. Enter the WHERE clause statement in the space provided. For example: WHERE email='clive@company.com'.

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 **Web SSO** in the steps list.
3. Click the **Configure Web SSO** button on the Web SSO screen.
4. Click **Adapter Mapping & User Lookup** in the steps list.
5. Click the adapter name.
6. Click **Database Filter** in the steps list.

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 232.

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 230).

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 [“Integration Kits and Adapters”](#) on page 39).

SAML2.0 Configuring 'saml2' IdP Connection [Help](#) | [Support](#) | [About](#) | [Logout \(Administrator\)](#)

[Main](#) | [IdP Connection](#) | [IdP Web SSO](#) | [Adapter Mapping & User Lookup](#)

[✓ Adapter Instance](#) | [✓ Adapter Data Store](#) | [✓ Data Store](#) | [✱ LDAP Directory Search](#) | [LDAP Filter](#) | [Adapter Contract Fulfillment](#) | [✓ Summary](#)

Please configure your directory search. This information, along with the attributes supplied in the SAML assertion, will be used to fulfill the SP Adapter Contract.

Base DN

Search Scope

Attributes to return from search

Root Object Class	Attribute	Action
	Subject DN	
- SELECT -		<input type="button" value="Add Attribute"/>

[View Adapter Contract](#)

Field Descriptions

Field	Definition
Base DN	The base distinguished name of where the beginning of the tree structure search begins. Searches look for information at or below this node.
Search Scope	Specifies the node depth of the query, which begins at the Base DN. Select Subtree, One level or Object.
Root Object Class	Specifies the object type within the LDAP hierarchy from which attributes will be returned.
Add Attribute	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.

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 **Web SSO** in the steps list.
3. Click the **Configure Web SSO** button on the Web SSO screen.
4. Click **Adapter Mapping & User Lookup** in the steps list.
5. Click the adapter name.

6. Click **LDAP Directory Search** in the steps list.

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 232).

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 you selected 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 161).

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”** to view a list from which to 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, `${username}` returns the value of `username` in the assertion.



Important: You *must* use the `${ }` syntax to retrieve the value of the enclosed variable.

Field Descriptions

Field	Definition
Filter	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.

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 **Web SSO** in the steps list.
3. Click the **Configure Web SSO** button on the Web SSO screen.
4. Click **Adapter Mapping & User Lookup** in the steps list.
5. Click the adapter name.
6. Click **LDAP Filter** in the steps list.

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 232).

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 82).

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 230). 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 required by the adapter from one of these Sources:

- Account Link

This source appears only if you have elected to use [account linking](#) (see [“Selecting Identity Mapping”](#) on page 225). 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 226).

- Expression

Values are derived from an expression written in the Object-Graph Navigation Language (OGNL), which is based on the Java programming language. OGNL expressions are useful for evaluating attribute values and returning information based on those values. 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("net-worth"),
#result=#result.replace("$",""),
```

```
#result=#result.replace(",",""),  
#result < 500000 ? "bronze" :  
#result < 1000000 ? "silver" : "gold"
```

You reference OGNL variables using the # symbol. PingFederate provides predefined OGNL variables for attributes received in an assertion. For example, the SAML_SUBJECT value may be retrieved using:

```
#SAML_SUBJECT
```

For data-store attributes, use this syntax (for example):

```
#this.get("ds.amount")
```

For attributes that contain a space or a special character, use this syntax (for example):

```
#this.get("net-worth")
```

For more information, see [“Using the OGNL Edit Screen”](#) on page 243.

For more information about OGNL, see the OGNL Web site, www.ognl.org.



Note: The PingFederate runtime engine uses OGNL version 2.6.7.

- 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 233, [“Configuring an LDAP Directory Search”](#) on page 236, or [“Selecting Custom Source Fields”](#) on page 240).

- 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.

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.

SAML2.0 Configuring 'saml2' IdP Connection [Help](#) | [Support](#) | [About](#) | [Logout \(Administrator\)](#)

[Main](#) | [IdP Connection](#) | [IdP Web SSO](#) | [Adapter Mapping & User Lookup](#)

✓ [Adapter Instance](#) | ✓ [Adapter Data Store](#) | ✓ [Data Store](#) | ✓ [LDAP Directory Search](#) | ✓ [LDAP Filter](#) | ✱ [Adapter Contract Fulfillment](#) | ✓ [Summary](#)

You can fulfill your Adapter Contract session-creation requirements with values from the assertion, hard-coded text, or from a data store lookup.

Adapter Contract	Source	Value
email	Assertion	email
userId	LDAP	Subject DN

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 **Web SSO** in the steps list.
3. Click the **Configure Web SSO** button on the Web SSO screen.
4. Click **Adapter Mapping & User Lookup** in the steps list.
5. Click the Adapter Instance Name.
6. Click **Adapter Contract Fulfillment** in the steps list.

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.
If you want to check for errors or test potential results of an OGNL expression, click **Edit** under Actions. For more information see [“Using the OGNL Edit Screen”](#) on page 243.
All values must be mapped.
3. Click **Done**.

Using the OGNL Edit Screen

An inline editor is available for OGNL expressions. The editor checks for errors in an expression and allows you to enter input values and test the resulting output.

- To reach the OGNL editor, click **Edit** under Actions for the expression.



Important: If you make changes to the expression and want to save them, click **Update** under Actions. To discard changes, click the **Cancel** link under Actions; click the **Cancel** button near the bottom of the screen only if you wish to discard all changes you have made in the current task.

To test an expression:

1. Enter an input value in the Source textbox 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.

Using the Adapter Mapping Summary Screen

When you have finished creating or modifying Adapter Mapping and User Lookup information, you can review the configuration on the Adapter Mapping 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 configuration, click **Done**; if you want to keep your changes, click **Save** when you reach the Web SSO 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 223).

SAML2.0 Configuring 'Partner1:entityId' IdP Connection [Help](#) | [Support](#) | [About](#) | [Logout \(Administrator\)](#)

[Main](#) | [IdP Connection](#) | [IdP Web SSO](#)

[✓ Identity Mapping](#) | [✓ Attribute Contract](#) | [✓ Adapter Mapping & User Lookup](#) | [✖ SSO Service URLs](#) | [✓ SLO Service URLs](#) | [✓ Allowable SAML Bindings](#) | [✓ Artifact Resolver Locations](#) | [✓ Signature Policy](#) | [✓ Encryption Policy](#) | [✓ Summary](#)

 As the SP, you send authentication requests (AuthnRequests) for single sign-on to the IdP's SSO Service. Depending on the situation, the IdP may have several endpoints available. Please provide the endpoints that you want to use when sending these requests.

Binding	Endpoint URL	Action
POST	/idp/SSO.saml2	Edit / Delete
Redirect	/idp/SSO.saml2	Edit / Delete
- SELECT - <input type="button" value="v"/> *	<input type="text"/> *	<input type="button" value="Add"/>

Field Descriptions

Field	Definition
Binding (SAML 2.0)	The transport type agreed upon by you and your partner: Artifact, POST, or Redirect.
Endpoint URL	The location where your IdP receives SSO messages.

To reach this screen:

- Click a connection name on the Main Menu.
Click **Manage All IdP**, if needed, to see a full list of connections.
- Click **Web SSO** in the steps list.
- Click **Configure Web SSO**.
- Click **SSO Service URLs** in the steps list.
If this step is not displayed, you have not selected SP-initiated SSO (see [“Choosing SAML Profiles”](#) on page 220).

To define an Endpoint URL:

- If you are using SAML 2.0, select the Binding your partner specifies for the Endpoint.
- Enter the fully qualified Endpoint URL or a relative path if you have defined a base URL (see [“General Information”](#) on page 217).
- 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.

WS-Fed Configuring 'wsf' SP Connection [Help](#) | [Support](#) | [About](#) | [Logout \(Administrator\)](#)

[Main](#) | [SP Connection](#) | [SP Web SSO](#)

[Identity Mapping](#) | [Attribute Contract](#) | [IdP Adapter Mapping](#) | [Service URL](#) | [Summary](#)

As the IdP, you send SAML assertions and SLO cleanup messages to the SP. Specify here the URL where the SP is expecting to receive these messages.

Endpoint URL *

- Enter the fully qualified URL or a relative path if you have defined a base URL (see [“General Information”](#) on page 217).

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 **Web SSO** in the steps list.
3. Click the **Configure Web SSO** button on the Web SSO screen.
4. Click **Service URL** in the steps list.

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 select either SLO profile (see [“Configuring IdP-Initiated SLO”](#) on page 223 or [“Configuring SP-Initiated SLO”](#) on page 224).

SAML2.0
Configuring 'Partner1:entityId'
IdP Connection
[Help](#) | [Support](#) | [About](#) | [Logout \(Administrator\)](#)

Main
IdP Connection
IdP Web SSO

Identity Mapping | Attribute Contract | Adapter Mapping & User Lookup | SSO Service URLs | SLO Service URLs | Allowable SAML Bindings | Artifact Resolver Locations | Signature Policy | Encryption Policy | Summary

As the SP, you may send SAML logout messages to the IdP's **Single Logout Service**. Depending on the situation, the IdP may have several endpoints available for this purpose. Please provide the endpoints you would like to use.

Binding	Endpoint URL	Response URL	Action
Redirect	/idp/SLO.saml2		Edit / Delete
POST	/idp/SLO.saml2		Edit / Delete
- SELECT -			Add

Field Descriptions

Field	Definition
Binding	The transport type agreed upon by you and your partner: Artifact, POST, Redirect, or SOAP.
Endpoint URL	The location where your IdP receives SLO request messages.
Response URL	(Optional) The location where the IdP receives logout responses. Use this endpoint when you are part of a chain of session participants.

To reach this screen:

- Click a connection name on the Main Menu.
Click **Manage All IdP**, if needed, to see a full list of connections.
- Click **Web SSO** in the steps list.
- Click **Configure Web SSO**.
- Click SLO Service in the steps list.

To define an Endpoint URL:

- Select the Binding your partner specifies for the Endpoint.
- Enter the fully qualified Endpoint URL or a relative path if you have defined a base URL (see [“General Information”](#) on page 217).
- (Optional) Enter the Response URL or a relative path and click **Add**.
- 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 220). You and your partner can agree to standardize on one binding type or select different bindings for different profile scenarios.

The screenshot shows the 'Configuring 'saml2' IdP Connection' window. The 'IdP Web SSO' tab is selected. In the steps list, 'Allowable SAML Bindings' is highlighted. The question 'When the IdP sends messages, over what SAML bindings do you want to receive them?' is displayed. Below it, four options are listed: 'Artifact' (unchecked), 'POST' (checked), 'Redirect' (unchecked), and 'SOAP' (checked).

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 **Web SSO** in the steps list.
3. Click **Configure Web SSO**.
4. Click **Allowable SAML Bindings** in the steps list.

To 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.

SAML2.0 Configuring 'Partner1:entityId' IdP Connection [Help](#) | [Support](#) | [About](#) | [Logout \(Administrator\)](#)

[Main](#) | **IdP Connection** | IdP Web SSO

[✓ Identity Mapping](#) | [✓ Attribute Contract](#) | [✓ Adapter Mapping & User Lookup](#) | [✓ SSO Service URLs](#) | [✓ SLO Service URLs](#) | [✓ Allowable SAML Bindings](#) | *** Artifact Lifetime** | [✓ Artifact Resolver Locations](#) | [✓ Signature Policy](#) | [✓ Encryption Policy](#) | [✓ Summary](#)

☐ Artifacts are meant to be short-lived tokens representing an issued message. For how long should the recipient of the artifact be allowed to retrieve the corresponding message?

Artifact Lifetime second(s) *

The default value is 60 seconds. You can change this 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:

1. Click a connection name on the Main Menu.
Click **Manage All IdP**, if needed, to see a full list of connections.
2. Click **Web SSO** in the steps list.
3. Click **Configure Web SSO**.
4. Click **Artifact Lifetime** in the steps list.
This step appears only if you have selected the Artifact binding for either a Web 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; SAML 1.x provides for a single endpoint.

SAML2.0 Configuring 'saml2' IdP Connection [Help](#) | [Support](#) | [About](#) | [Logout \(Administrator\)](#)

[Main](#) | **IdP Connection** | IdP Web SSO

[✓ Identity Mapping](#) | [✓ Attribute Contract](#) | [✓ Adapter Mapping & User Lookup](#) | [✓ SSO Service URLs](#) | [✓ SLO Service URLs](#) | [✓ Allowable SAML Bindings](#) | *** Artifact Resolver Locations** | [✓ Signature Policy](#) | [✓ Encryption Policy](#) | [✓ Summary](#)

☐ Please provide the remote party URLs that you will use to resolve/translate the artifact and get the actual protocol message.

Index	URL	Action
<input type="text"/>	<input type="text"/>	<input type="button" value="Add Artifact Resolution Service"/>

Note that this screen is not the same for SAML 1.x implementations.

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 **Web SSO** in the steps list.
3. Click **Configure Web SSO**.
4. Click **Artifact Resolver Locations** in the steps list.
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 Artifact Resolution Service**.

The URL must be fully qualified (defining protocol, host, and port) unless you have entered a base URL (see [“General Information”](#) on page 217).

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 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 217).

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 their Issuer ID, then enter the Source ID supplied by your IdP partner.

3. Click **Next**.

Configuring Signature Policy

The Signature Policy screen provides optional settings for digital signatures. The choices you make on this screen depend on your partner agreement (see [“Digital Signing Policy Coordination”](#) on page 44).

Digital signing is required for SAML Response messages, including assertions, from the IdP via POST (or Redirect for SAML 2.0). Optionally, SSO authentication requests from the SP (SP-initiated SSO) may be signed under SAML 2.0 specifications. Check this option if your partner agreement includes this option. (The option appears only if you have enabled SP-initiated SSO using the POST or redirect bindings.)

Assertions inside SAML Responses may also be separately signed. If your partner agreement includes this option, click the relevant checkbox on this screen. (This is the only choice on the SAML 1.x screen.)

SAML2.0 Configuring 'saml2' IdP Connection [Help](#) | [Support](#) | [About](#) | [Logout \(Administrator\)](#)

[Main](#) | **IdP Connection** | IdP Web SSO

✓ Identity Mapping | ✓ Attribute Contract | ✓ Adapter Mapping & User Lookup | ✓ SSO Service URLs | ✓ SLO Service URLs | ✓ Allowable SAML Bindings | ✖ **Signature Policy** | ✓ Encryption Policy | ✓ Summary

Additional guarantees of authenticity may be agreed upon between you and your partner. For SP-initiated SSO, you can choose to sign authentication requests sent via the POST or redirect bindings. You can also choose to require signed assertions, regardless of the binding used.

☐ Sign AuthN requests over POST and Redirect bindings

☐ Require digitally signed SAML Assertion

- Make your choices 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 that profile).

Note that "The entire assertion" selection on the Encryption Policy screen includes the SAML_SUBJECT and all attributes.

SAML2.0 Configuring 'saml2' IdP Connection [Help](#) | [Support](#) | [About](#) | [Logout \(Administrator\)](#)

[Main](#) | [IdP Connection](#) | [IdP Web SSO](#)

[Identity Mapping](#) | [Attribute Contract](#) | [Adapter Mapping & User Lookup](#) | [SSO Service URLs](#) | [SLO Service URLs](#) | [Allowable SAML Bindings](#) | [Signature Policy](#) | **Encryption Policy** | [Summary](#)

Additional guarantees of message level privacy may be used between you and your partner through the use of XML encryption. Specify an encryption policy for the exchange of SAML messages.

☐ None
☒ Encryption enabled

☒ The entire assertion
☐ SAML_SUBJECT (Name Identifier)
☐ One or more attributes

☒ Encrypt the SAML Subject in SLO messages to the IdP
☒ Require that the SAML Subject be encrypted in SLO messages from the IdP

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 **Web SSO** in the steps list.
3. Click the **Configure Web SSO** button on the Web SSO screen.
4. Click **Encryption Policy** in the steps list.

To define XML encryption:

1. Select Encryption enabled.
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 require 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 Web SSO screen.

Editing and Saving Web SSO Configurations

On the Summary screen you can review or edit your Web SSO configuration.



Important: When you finish editing existing profiles, you must click **Done** on the Summary screen and then **Save** on the Web SSO screen. For a new connection, click **Done** and then click **Next** on the Web SSO screen. Save the entire connection on the Activation screen (see [“Connection Activation and Summary”](#) on page 264).

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 dependent or other changes, do so and continue by clicking **Done** until you reach the Web SSO screen.

3. Click **Save** on the Web SSO screen.

Configuring the Attribute Query Profile

At the Attribute Query step you configure your connection to request attributes from your partner IdP, if you have chosen this profile (see [“Choosing SAML Profiles”](#) on page 220). Attribute queries are not dependent on single sign-on but may be used independently or in conjunction with Web SSO to provide flexibility in how a user authenticates with SP applications (see [“Attribute Query and XASP”](#) on page 31).

Setting the Attribute Authority Service URL

An *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 Attribute Authority partner (see [“Attribute Query and XASP”](#) on page 31).

The screenshot shows a web interface for configuring a SAML2.0 connection. The title bar says 'SAML2.0 Configuring Partner1:entityId IdP Connection' with links for Help, Support, About, and Logout (Administrator). Below the title bar are tabs: Main, IdP Connection, and Attribute Query (which is selected). Under the Attribute Query tab, there are four sub-sections: Attribute Request Service URL (with a star icon), Attribute Name Mapping (with a checkmark), Security Policy (with a checkmark), and Summary (with a checkmark). The Attribute Request Service URL section is highlighted in green and contains the text 'Specify the URL at your IdP partner's site where attribute queries are to be sent.' Below this text is a text input field labeled 'Attribute Authority Service URL' containing the value '/idp/attrsvc.ssaml2' followed by an asterisk.

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 217).

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.

The screenshot shows a web interface for configuring a SAML2.0 connection. The title bar says 'SAML2.0 Configuring' and 'Doc_working_from_Devbuild' IdP Connection. There are links for Help, Support, About, and Logout (Administrator). The main navigation bar has tabs for Main, IdP Connection, Attribute Authority, and a highlighted tab. Below the navigation bar, there are links for Attribute Request Service URL, Attribute Name Mapping (which is selected), Security Policy, and Summary. A green box contains a warning: 'Attributes requested by your application may not match exactly the attributes supplied by the IdP. Specify here the mapping between these sets of attributes.' Below this is a table with three columns: Local Name, Remote Name, and Action. The table contains four rows of mappings: EmailAddress to email, userid to SAML_SUBJECT, FirstName to fname, and LastName to lname. Each row has an 'Edit / Delete' link in the Action column. At the bottom of the table, there are input fields for Local Name and Remote Name, and an 'Add' button.

Local Name	Remote Name	Action
EmailAddress	email	Edit / Delete
userid	SAML_SUBJECT	Edit / Delete
FirstName	fname	Edit / Delete
LastName	lname	Edit / Delete
<input type="text"/>	<input type="text"/>	<input type="button" value="Add"/>

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.

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 255).

This screen also allows you to mask incoming attribute values in log files (see “Attribute Masking” on page 42). When you enable this selection, all user attributes returned from this IdP are masked.

SAML2.0 Configuring 'saml2' IdP Connection [Help](#) | [Support](#) | [About](#) | [Logout \(Administrator\)](#)

[Main](#) | [IdP Connection](#) | [Attribute Query](#) | [Security Policy](#)

✓ Attribute Request Service URL | ✓ Attribute Name Mapping | ✖ **Security Policy** | ✓ Summary

Specify the attribute authority profile's security policy with your partner.

- ☒ Require signed Response
- ☒ Require signed Assertion
- ☒ Require encrypted assertion
- ☐ Sign the Attribute Query
- ☐ Encrypt the Name Identifier
- ☒ Mask attributes in log files

To configure attribute-query security policy for this partner:

- Check or uncheck the 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 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 SAML Web SSO or Attribute Query configuration may involve any or all of the following:

- [Configuring Back-Channel Authentication](#)
- [Configuring Digital Signature Settings](#)
- [Selecting Signature Verification Certificates](#)
- [Selecting an Encryption Certificate](#)
- [Selecting a Decryption Key](#)

SAML2.0 Configuring 'Partner1:entityId' IdP Connection [Help](#) | [Support](#) | [About](#) | [Logout \(Administrator\)](#)

[Main](#) | **IdP Connection**

✓ [Role & Protocol](#) | ✓ [General Info](#) | ✓ [SAML Profiles](#) | ✓ [Web SSO](#) | ✓ [Attribute Query](#) | *** [Credentials](#)** | ✓ [Activation & Summary](#)

For each credential shown here, configure the necessary settings.

Credential	
Back-Channel Authentication	Configured
Digital Signature	Configured
Digital Signature Verification	Configured
Encryption Certificate	Configured
Decryption Certificate	Configured

[Configure Credentials](#)

- To begin configuring credentials, click the **Configure Credentials** button on the Credentials screen.

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 220).

SAML2.0
Configuring 'Partner1:entityId' IdP Connection
[Help](#) | [Support](#) | [About](#) | [Logout \(Administrator\)](#)

[Main](#)
[IdP Connection](#)
[Credentials](#)

✱ **Back-Channel Authentication** | ✓ Digital Signature Settings | ✓ Signature Verification Certificate | ✓ Select XML Encryption Certificate | ✓ Select XML Decryption Key | ✓ Summary

 You selected one or more bindings that require additional security for communication with your partner. Please ensure that security settings are properly configured.

Send to your partner: <ul style="list-style-type: none"> • SOAP SLO messages • Artifact resolution requests • Attribute Query requests 	Configure
Receive from your partner: <ul style="list-style-type: none"> • SOAP SLO messages • Artifact resolution requests • Attribute Query requests 	Configure



Note: A yellow triangle next to a listing indicates that you have not completely configured back-channel authentication requirements.

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 **Credentials** in the steps list.
3. Click the **Configure Credentials** button.

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 the SOAP Authentication Type on the next screen:
 - Basic — you will enter SOAP Basic credentials on a later screen.
 - SSL Certificate — you will specify the certificate on a later screen.

This option is enabled only if you have specified an endpoint that uses SSL.

You and your partner might also have agreed to require a valid certificate chain; if so, select the checkbox for that option.

- Use Digital Signatures . . . — you will sign the message.

You will be asked to select a signing certificate on a later screen.

All three options may be combined or used separately.

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 115). You will need to export the certificate (only) and send it 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. Click **Next**.

3. 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.

All three options may be combined or used separately.

Click the checkbox to Require SSL if the connection will use the secure protocol for basic or digital signature authentication.

4. Click **Next**.

5. If you chose Basic at Step 3, 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.

6. 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 CAs”](#) on page 111).
 - 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.

7. Click **Next**.
8. If you chose anchored SSL certificate verification at [Step 6](#), 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.

9. If you chose unanchored SSL certificate verification at [Step 6](#), select the certificate you will use to validate the SSL connection.

If you have not yet imported the certificate into PingFederate, click **Manage Certificates**.

10. Click **Next**.
11. 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 220) using the POST or Redirect binding. 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 255).

The step is not required for SAML 1.x IdP connections.

SAML2.0 Configuring 'saml2' IdP Connection [Help](#) | [Support](#) | [About](#) | [Logout \(Administrator\)](#)

Main | **IdP Connection** | Credentials

* **Digital Signature Settings** | ✓ Signature Verification Certificate | ✓ Summary

You may need to digitally sign SAML messages sent to your partner to protect against tampering. Please select a key/certificate to use from the list below.

1172618811984 (cn=test) *

Manage Certificates...

☐ Include Key Info with the SAML message

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 **Credentials** in the steps list.
3. Click the **Configure Credentials** button.
4. Click **Digital Signature Settings** in the steps list.

If this step does not appear, then your configuration does not require digital signatures. You do not have SLO configured using the POST or Redirect binding, and you have not elected to sign either authentication or attribute requests (see [“Configuring Signature Policy”](#) on page 250 and [“Specifying Security Policy”](#) on page 255).

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 118).
2. (Optional) If you have agreed to send your public key with the SAML message, click the checkbox to implement this requirement.

Selecting Signature Verification Certificates

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.

Depending on your agreement with this IdP, SSO assertions, SAML 2.0 artifacts, or SOAP messages might also require signatures.

Whenever signatures are required, you must import your partner's public key certificate into the PingFederate store for signature verification.



Tip: To prevent any interruption of service due to an expired certificate, you can ask your partner for a new certificate in advance and use it in the Secondary certificate field. The PingFederate server will use the primary certificate until it expires and then try the secondary.

The screenshot shows the 'Configuring 'saml2' IdP Connection' page. The 'IdP Connection' tab is selected, and the 'Signature Verification Certificate' sub-tab is active. A green box contains instructions: 'Incoming SAML messages may be signed by your partner. Please select which certificate(s) to use when verifying these digital signatures. When multiple certificates are chosen, each certificate is tried from the top of the list down until the signature is verified.' Below this, there are two dropdown menus labeled 'Primary' and 'Secondary', both currently set to '- SELECT -'. A 'Manage Certificates...' button is at the bottom.

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 **Credentials** in the steps list.
3. Click the **Configure Credentials** button.
4. Click **Signature Verification Certificate** in the steps list.

If this step does not appear for SAML 2.0, then your configuration does not require a verification certificate. You are not using SLO POST or Redirect bindings, and signed assertions are not required (see “[Choosing Allowable SAML Bindings \(SAML\)](#)” on page 248 and “[Configuring Signature Policy](#)” on page 250).

To specify a certificate:

1. Select the certificate from the drop-down list.
If you have not yet imported the certificate into PingFederate, click **Manage Certificates**.
2. Optionally, 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 251.

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 255).

The screenshot shows a web interface for configuring a SAML2.0 IdP connection. The title bar says 'SAML2.0 Configuring 'saml2' IdP Connection' with links for Help, Support, About, and Logout (Administrator). Below the title bar are tabs for Main, IdP Connection, and Credentials. The IdP Connection tab is active, showing a list of steps: Digital Signature Settings, Signature Verification Certificate, Select XML Encryption Certificate (highlighted), Select XML Decryption Key, and Summary. A green box contains the instruction: 'Please select the partner certificate to use when encrypting message content as well as the preferred block encryption and key transport algorithms. Only RSA keys can be used for XML encryption.' Below this are two sections: 'Block Encryption Algorithm' with radio buttons for AES-128 (selected), AES-256 (with a help link), and Triple DES; and 'Key Transport Algorithm' with radio buttons for RSA-v1.5 (selected) and RSA-OAEP. At the bottom is a dropdown menu labeled '- SELECT -' and a button labeled 'Manage Certificates...'.

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 **Credentials** in the steps list.
3. Click the **Configure Credentials** button.
4. Click **Select XML Encryption Certificate** in the steps list.

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 220) or you have chosen not to encrypt the assertion or the `SAML_SUBJECT` (see [“Configuring XML Encryption Policy \(SAML 2.0\)”](#) on page 251).

To identify the encryption certificate:

1. (Optional) Change the default settings under Block Encryption Algorithm and/or Key Transport Algorithm.

Note that the use of stronger AES encryption is subject to export control restrictions. The standard JRE distribution does not support this encryption.

To use the strongest AES encryption, when permissible, download and install the “JCE [Java Cryptography Extension] Unlimited Strength Jurisdiction Policy Files” from <http://java.sun.com/products/jce/javase.html#UnlimitedDownload>.

For more information about XML block encryption and key transport algorithms, see the “[XML Encryption Syntax and ProcessingW3C Recommendation](http://www.w3.org/TR/xmlenc-core/)” at <http://www.w3.org/TR/xmlenc-core/>.

- 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 251).

To reach this screen:

- Click a connection name on the Main Menu.
Click **Manage All IdP**, if needed, to see a full list of connections.
- Click **Credentials** in the steps list.
- Click the **Configure Credentials** button.
- 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 251).

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 118).



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. No messages are actually sent or received until your partner’s federation gateway is also established and a user actually initiates an SSO or SLO event.



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.

To change a Connection Status:

- Select either Active or Inactive and then click **Save**.



Important: Be sure to click **Save**. Otherwise, the status will not be changed.

To modify a connection:

1. Click the heading above the information you want to change.
2. Change the information on the step screen and click **Done** or **Save**.

3. Change any dependent information on other screens if needed.
PingFederate identifies dependencies for you.
4. When you return to the Activation & Summary screen or to a task summary screen, click **Save**.



Important: Be sure to click **Save**. Otherwise, the connection will not be reconfigured.

Standard Adapter Configuration

In order to transfer identity and other user-attribute information between the PingFederate server and an end application, the product architecture allows for custom adapters to be deployed with the server (see “[Integration Kits and Adapters](#)” on page 39).

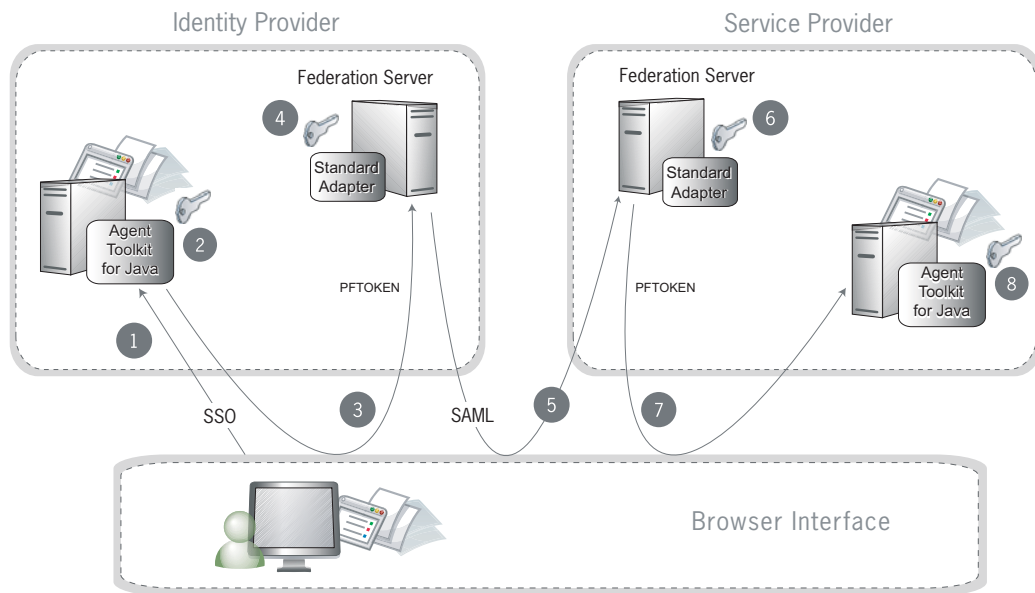
PingFederate ships with a predeployed “Standard Adapter.” The Standard Adapter uses a proprietary, secure token format (PFTOKEN) to transfer user attributes between an application and the PingFederate server. On the IdP side, the Standard Adapter allows the PingFederate server to receive a user's identity from the IdP application. On the SP side, the Standard Adapter can be used to transfer user identity information to the target SP application.

Specialized application integration kits (for example, the Java Integration Kit and the .NET Integration Kit) are available from www.pingidentity.com. Each kit leverages the Standard Adapter, in concert with an “Agent Toolkit,” to integrate applications with the PingFederate server. The agent portion of the integration kits resides with the application and uses PFTOKEN to communicate with the Standard Adapter.



Note: To integrate applications for use with the Standard 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 Standard 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.



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 a PFTOKEN.
Attributes are encrypted and decrypted using the Java Cryptography Extension (JCE). For more information, see <http://java.sun.com/products/jce>.
3. A request containing the PFTOKEN is redirected to the PingFederate IdP server.
4. The server invokes the Standard IdP Adapter, which retrieves the PFTOKEN, 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 Standard SP Adapter. The Adapter encrypts the data internally and generates a PFTOKEN.
7. A request containing the PFTOKEN is redirected to the SP application.
8. The Agent Toolkit for Java decrypts and parses the PFTOKEN and makes the attributes available to the SP Application.

Configuring the IdP Standard Adapter

1. If you have not already done so, log on to the PingFederate administrative console and click **IdP Adapters** on the Main Menu.
2. On the Manage Adapter Instances screen, click **Create New Adapter Instance**.
3. On the Adapter Type screen, enter an Adapter Instance Name and Adapter Instance Id, select PF4 Standard Adapter v1.1 as the Adapter Type, and click **Next**.

The Adapter 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.



Note: If you do not know the values to enter at this time, use placeholders, following formats as shown in the figure below. If you are new to PingFederate, you might wish to follow an example setup in the `quickstart/docs/Quick_Start_Guide.pdf` to configure a sample application. You can return to this screen to enter correct values for a real connection at any time before deployment (see “[Configuring IdP Adapters](#)” on page 124).

Configuring IdP Adapter
[Help](#) | [Support](#) | [About](#) | [Logout \(Administrator\)](#)

[Main](#) | [Manage IdP Adapter Instances](#) | [Create Adapter Instance](#)

[Adapter Type](#) | [IdP Adapter](#) | [Adapter Actions](#) | [Extended Adapter Contract](#) | [Adapter Attributes](#) | [Summary](#)

Complete the configuration necessary to look up user security contexts in your environment. This configuration was designed into the adapter for use at your site.

This is the standard adapter for PingFederate. The adapter uses a proprietary, secure token format (PFTOKEN) to transfer attributes between an application and the PingFederate server.

Field Name	Field Value	Description
Transfer method	<input type="radio"/> Cookie <input checked="" type="radio"/> Query parameter	How the PFTOKEN is transferred, either via a cookie or as a query parameter.
PFTOKEN holder name	IdPJava	The name of the cookie, or the query parameter that contains the PFTOKEN. This name should be unique for each adapter instance.
Domain	.pingidentity.com	The server domain, preceded by a period (e.g., .pingidentity.com). If no domain is specified, the value is obtained from the request
Cookie path	/	The path for the cookie that contains the PFTOKEN.
Encode Cookie	<input type="checkbox"/>	If checked, the PFTOKEN cookie value is encoded to allow interoperability with various application servers.
Password	*****	The password used for generating a key to encrypt data.
Logout Service		The URL to which the user is redirected for a Single Logout (SLO) event. This URL is part of an external application, which terminates the user session.
Authentication Service	http://localhost:8080/IdpSample?c	The URL to which the user is redirected for an SSO event. This URL is part of an external application, which performs user authentication.

Show Advanced Fields

5. Click **Next**.



Note: See “Configuring Advanced Fields” on page 276 for information on additional, optional settings.

6. On the Adapter Actions screen, click **Generate properties**.

Action Name	Action Description	Action Invocation Link
Generate properties	Generate properties for the agent side	Invoke Generate properties

7. On the next screen, click **Export** and save the properties file.
- The values in the resulting file `pfagent.properties` are established by the console configuration and are used by the IdP application. Refer to either the Java or .NET Integration Kit *User Guide* for details.
8. (Optional) On the Extended Adapter Contract screen, you can configure additional attributes for the adapter (see “[Extending an Adapter Contract](#)” on page 127).

Adapter Contract
userId

Extend the Contract	Action
<input type="text"/>	<input type="button" value="Add"/>

9. Click **Next**.
10. On the Adapter Attributes screen, select the `userId` checkbox under Pseudonym (optionally, select other attributes, if you added any attributes at [Step 8](#)).

Attribute	Pseudonym	Mask Log Values
userId	<input type="checkbox"/>	<input type="checkbox"/>

This selection is used if any of your SP partners will make use of [pseudonyms](#) for [account linking](#) (see “[Account Linking](#)” on page 38).



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.

As an option on this screen, you can choose to mask the values of any or all attributes from the adapter that PingFederate logs at runtime (see “[Attribute Masking](#)” on page 42).

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.

Configuring the SP Standard Adapter

1. If you have not already done so, log on to the PingFederate administrative console and click **SP Adapters** on the Main Menu.
2. On the Manage Adapter Instances screen, click **Create New Adapter Instance**.
3. Enter an Adapter Instance Name and Adapter Instance Id, select PF4 Standard Adapter v1.1 as the Adapter Type, and click **Next**.

The Adapter Instance Id may not contain spaces or underscores.

Configuring 'SPJava' SP Adapter		Help Support About Logout (Administrator)
Main	Manage SP Adapter Instances	Create Adapter Instance
* Adapter Type SP Adapter Instance Adapter Actions Extended Adapter Contract Summary		
<div>Please enter an Adapter Instance Name and Id, and select the Adapter Type.</div>		
Adapter Instance Name	<input type="text" value="SPJava"/>	*
Adapter Instance Id	<input type="text" value="SPJava"/>	*
Adapter Type	<input type="text" value="PF4 Standard Adapter v1.2"/>	* Visit PingIdentity.com for additional adapter types

4. Enter values for the adapter configuration on the SP Adapter Instance screen.

These values are dependent on your developer's implementation.



Note: If you do not know the values to enter at this time, use placeholders, following formats as shown in the figure below. If you are new to PingFederate, you might wish to follow an example setup in the `quickstart/docs/Quick_Start_Guide.pdf` to configure a sample application. You can return to this screen to enter the correct values for your deployment at any time (see ["Configuring SP Adapters"](#) on page 198).

Configuring 'asd' SP Adapter
[Help](#) | [Support](#) | [About](#) | [Logout \(Administrator\)](#)

[Main](#) | [Manage SP Adapter Instances](#) | [Create Adapter Instance](#)

✓ Adapter Type | ✖ **SP Adapter Instance** | [Adapter Actions](#) | [Extended Adapter Contract](#) | [Summary](#)

Complete the configuration necessary to set the appropriate security context for user sessions in your environment. This configuration was designed into the adapter for use at your site.

This is the standard adapter for PingFederate. The adapter uses a proprietary, secure token format (PFTOKEN) to transfer attributes between an application and the PingFederate server.

Field Name	Field Value	Description
Transfer method	<input type="radio"/> Cookie <input checked="" type="radio"/> Query parameter	How the PFTOKEN is transferred, either via a cookie or as a query parameter.
PFTOKEN holder name	<input type="text" value="SPJava"/>	The name of the cookie, or the query parameter that contains the PFTOKEN. This name should be unique for each adapter instance.
Domain	<input type="text" value=".pingidentity.com"/>	The server domain, preceded by a period (e.g., .pingidentity.com). If no domain is specified, the value is obtained from the request
Cookie path	<input type="text" value="/"/>	The path for the cookie that contains the PFTOKEN.
Encode Cookie	<input type="checkbox"/>	If checked, the PFTOKEN cookie value is encoded to allow interoperability with various application servers.
Password	<input type="password" value="*****"/>	The password used for generating a key to encrypt data.
Logout Service	<input type="text"/>	The URL to which the user is redirected for a Single Logout (SLO) event. This URL is part of an external application, which terminates the user session.
Authentication Service	<input type="text" value="http://localhost:8080/SpSample/?c"/>	The URL to which the user is redirected for an SSO event. This URL is part of an external application, which performs user authentication.
Account Link Service	<input type="text"/>	The URL to which the user is redirected for Account Linking. This URL is part of an external SP application, which performs user authentication and returns the local userid through PFTOKEN.

Show Advanced Fields

5. Click **Next**.

6. On the Adapter Actions screen, click **Generate properties**.

Action Name	Action Description	Action Invocation Link
Generate properties	Generate properties for the agent side	Invoke Generate properties

7. On the next screen, click **Export** and save the properties file.
- The values in the resulting file `pfaagent.properties` are established by the console configuration and are used by the SP application. Refer to either the Java or .NET Integration Kit *User Guide* for details.
8. Click **Next**.
9. (Optional) On the Extended Adapter Contract screen, you can configure additional attributes for the adapter (see “[Extending an Adapter Contract](#)” on page 127).
10. Click **Next**.
11. On the Summary screen, review the configuration and click **Done**.
- You can also click any heading to go back and change information.
12. 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 a Standard Adapter configuration, then you must first click **Next** and map target URLs to adapter instances (see “[Mapping URLs to SP Adapter Instances](#)” on page 205).

Configuring Advanced Fields

Advanced fields can be used to reconfigure default attributes for the PFTOKEN. To reach the Advanced Fields screen, click **Show Advanced Fields** on the IdP Adapter Instance or SP Adapter Instance screens.

Cookie max age	<input type="text" value="300"/> *	The max age (in seconds) of PFTOKEN. This value is also used as the max age of the cookie containing PFTOKEN.
Delete Cookie	<input checked="" type="checkbox"/>	If checked, the PFTOKEN cookie would be deleted after consumption. This option is valid only if the Transfer Method is 'Cookie'.
Algorithm	<input checked="" type="radio"/> AES <input type="radio"/> 3DES	The name of the encryption algorithm 3DES, or AES.
Mode	<input checked="" type="radio"/> CBC <input type="radio"/> ECB <input type="radio"/> CFB <input type="radio"/> OFB <input type="radio"/> PCBC	The cipher mode for encryption
Key size	<input checked="" type="radio"/> 128 <input type="radio"/> 192 <input type="radio"/> 256 <input type="radio"/> 56 <input type="radio"/> 112 <input type="radio"/> 168	The size of the encryption key in bits. Due to the import restrictions of some countries, the jurisdiction policy files distributed by default with Java(TM) have built-in restrictions on available cryptographic strength (key size). The use of certain size cryptographic keys requires that you have the 'Unlimited Strength Java(TM) Cryptographic Extension (JCE) Policy Files' installed in your JRE or JDK. See http://java.sun.com/ for download and installation instructions.
Iteration count	<input type="text" value="1000"/> *	The number of iterations for generating the encryption key
UserId Attribute Name	<input type="text" value="userId"/>	The name of the attribute which represents userId in PFTOKEN.

Field Descriptions

Field	Definition
Cookie max age	The max age (in seconds) of the cookie that contains PFTOKEN.
Delete cookie	If checked, the PFTOKEN cookie would be deleted after consumption. This option is valid only if the Transfer Method is 'Cookie'.
Algorithm	The name of the encryption algorithm—DES, 3DES, or AES.
Mode	Valid values are CBC, CFB, ECB, OFB, and PCBC. CBC mode is recommended. For details, see http://java.sun.com/j2se/1.5.0/docs/guide/security/jce/JCERefGuide.html#AppA . The Agent Toolkit for .NET supports only the default cipher modes CBC and ECB.

Field	Definition
Key size	The size of the key in bits. Values for the following algorithms are: AES – 128, 192, or 256 3DES – 112 or 192 DES – 56
Iteration count	The number of iterations for generating the client key. The default, 1000, is recommended.
UserId Attribute Name	The name of the attribute which represents userId in PFTOKEN.

Four additional fields are available for the SP adapter only:

Field	Definition
Iteration count	The number of iterations for generating the encryption key.
UserID by Query String	Checking this box will send the userId in clear along with PFTOKEN, as part of the query string. (This option is only applicable if the transfer method is Query Parameter).
User ID	The parameter name used for userId when it is being sent as part of the query string.
Send extended attributes	Select the method of sending extended attributes. These attributes are included in PFTOKEN and can be sent along with the request through request header or browser cookies.

Note that the use of AES encryption is subject to import control restrictions. The version of JCE policy files bundled in the J2EE 1.4.x and later environment allows for “strong” but limited cryptography. To use the strongest AES encryption, download and install the JCE “Unlimited Strength Jurisdiction Policy Files” from <http://java.sun.com/products/jce/javase.html#UnlimitedDownload>.

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. See, for example, any of the following sites:

- <http://www.securityfocus.com/infocus/1192>
- http://www.cert.org/tech_tips/unix_configuration_guidelines.html#A
- http://www.windowsecurity.com/articles/Passwords_Improve_Windows_Security_Part1.html

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 [“Integration Kits and Adapters”](#) on page 39).

PingFederate packages an LDAP Authentication Service 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 38).

PingFederate and the LDAP SP adapter handle account linking in the following way:

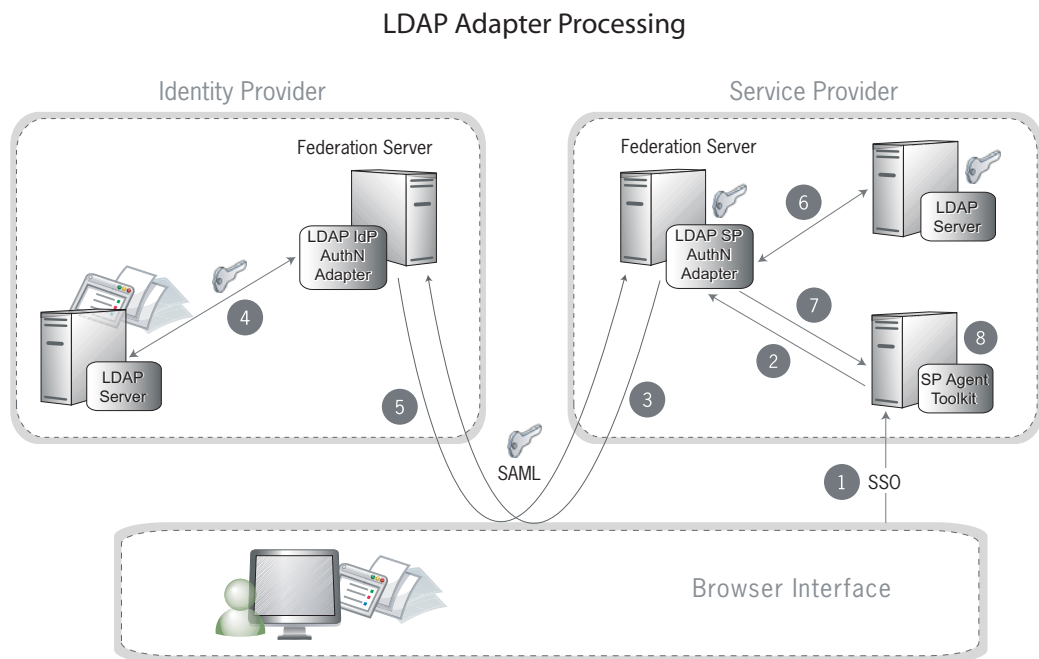
1. The adapter prompts the user for local SP credentials and validates the credentials against the data store.
2. The adapter passes the user ID to PingFederate to save in an embedded account-linking data store. (You can use the SDK to extend account linking to an external data store.)
3. The LDAP SP adapter then employs the PingFederate Java or .NET integration kit to transfer user attributes to the SP application (see [“Standard Adapter Configuration”](#) on page 267).

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. Several application-type-specific agent toolkits are available from www.pingidentity.com.



Note: To integrate SP applications for use with the LDAP Adapter, download an integration kit for PingFederate from www.pingidentity.com and follow instructions for installing and using 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 following figure shows a basic SP-initiated SSO scenario with PingFederate servers using the LDAP Authentication Service 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 a PFTOKEN.
7. A request containing the PFTOKEN is redirected to the SP application.
8. The SP Agent Toolkit decrypts and parses the PFTOKEN 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 82).
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 Adapter Instance Name and Adapter Instance Id, select PF4 LDAP Authentication Service 1.0 as the Adapter Type, and click **Next**.

The Adapter Instance Id may not contain spaces or underscores.

5. On the IdP Adapter screen, enter the values for adapter configuration described below.



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 124). Click **Manage Data Stores** if you have not established a connection to your LDAP server.

Configuring IdP Adapter
[Help](#) | [Support](#) | [About](#) | [Logout \(Administrator\)](#)

Main
Manage IdP Adapter Instances
Create Adapter Instance

✓ Adapter Type | ✖ IdP Adapter | ✓ Adapter Attributes | Summary

Complete the configuration necessary to look up user security contexts in your environment. This configuration was designed into the adapter for use at your site.

PF4 LDAP Authentication Service v1.0

Field Name	Field Value	Description
LDAP Datastore	-- Select One --	Select LDAP Datastore
Search Base	user	The location in the directory from which the LDAP search begins
Search Filter	\${username}	You may use \${username} and \${domain} to indicate user name and user domain accordingly
Realm	pingidentity.com	Authentication Realm: a name which is associated with the protected area

Manage Data Stores...
Show Advanced Fields

Property	Description
LDAP Datastore	The LDAP Data store configured in PingFederate.
Search Base	The location in the LDAP directory server from which the search begins.
Search Filter	A filter for entries in the directory used to produce the desired set of matching records. The search filter is a Boolean combination of attribute value assertions. You can use the <code>\${username}</code> and <code>\${domain}</code> variables, which represent the username and user domain entered by the user.
Realm	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.

6. (Optional) Click **Show Advanced Fields** and change parameters as needed.

Scope of Search	<input type="radio"/> One Level <input checked="" type="radio"/> Subtree	
Connection Pooling	<input type="checkbox"/>	
Operational Mode	<input checked="" type="radio"/> HTTP Basic Authentication <input type="radio"/> HTML Form Authentication	
Challenge Retries	<input type="text" value="3"/>	Max value of User Challenge Retries

Property	Description
Scope of Search	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.
Connection Pooling	A type of connection sharing supported by the LDAP server, which maintains a pool of (possibly) previously used connections and assigns them as needed.
Operational Mode	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.
Challenge Retries	The number of attempts allowed for password authentication.

7. Click **Next**.
8. On the Adapter Attributes screen, select the `userId` checkbox under Pseudonym (and, optionally, other attributes, if available).

Attribute	Pseudonym	Mask Log Values
userId	<input type="checkbox"/>	<input type="checkbox"/>

This selection is used if any of your SP partners will make use of [pseudonyms](#) for [account linking](#) (see “[Account Linking](#)” on page 38).



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.

Optionally on this screen, 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 42).

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.



Note: If this is the second instance of a Standard Adapter configuration, then you must first click **Next** and map target URLs to adapter instances (see “[Mapping URLs to SP Adapter Instances](#)” on page 205).

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 82).
2. Click **SP Adapters** on the Main Menu.

3. On the Manage Adapter Instances screen, click **Create New Adapter Instance**.
4. On the Adapter Type screen, enter an Adapter Instance Name and Adapter Instance Id, select PF4 LDAP Authentication Service 1.0 as the Adapter Type, and click **Next**.

The Adapter Instance Id may not contain spaces or underscores.

5. On the SP Adapter Instance 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 198). Click **Manage Data Stores** if you have not established a connection to your LDAP server.

Configuring 'LDAP One' SP Adapter
[Help](#) | [Support](#) | [About](#) | [Logout \(Administrator\)](#)

[Main](#) | [Local Settings](#) | [Manage SP Adapter Instances](#) | [Create Adapter Instance](#)

[Adapter Type](#) | [SP Adapter Instance](#) | [Adapter Actions](#) | [Extended Adapter Contract](#) | [Summary](#)

Complete the configuration necessary to set the appropriate security context for user sessions in your environment. This configuration was designed into the adapter for use at your site.

PF4 LDAP Authentication Service v1.0

Field Name	Field Value	Description
Logout Service	<input type="text" value="http://my_company:9030/"/> *	The URL to which the user is redirected for a Single Logout (SLO) event. This URL is part of an external application, which terminates the user session.
Transfer method	<input type="radio"/> Cookie <input checked="" type="radio"/> Query parameter	How the PFTOKEN is transferred, either via a cookie or as a query parameter.
PFTOKEN holder name	<input type="text" value="pftoken_holder"/> *	The name of the cookie, or the query parameter that contains the PFTOKEN. This name should be unique for each adapter instance.
Domain	<input type="text"/>	The server domain, preceded by a period (e.g., .pingidentity.com).
Cookie path	<input type="text" value="/"/> *	The path for the cookie that contains the PFTOKEN.
Password	<input type="password" value="....."/> *	The password used for generating a key to encrypt data.
Account Link Service	<input type="text"/>	The URL to which the user is redirected for Account Linking. This URL is part of an external SP application, which performs user authentication and returns the local userid through PFTOKEN.
Authentication Service	<input type="text"/>	The URL to which the user is redirected for an SSO event. This URL is part of an external application, which performs user authentication.
LDAP Datastore	-- Select One -- <input type="button" value="v"/> *	Select LDAP Datastore
Search Base	<input type="text" value="DN"/> *	The location in the directory from which the LDAP search begins
Search Filter	<input type="text" value="(userid=\${username})"/> *	You may use \${username} and \${domain} to indicate user name and user domain accordingly
Realm	<input type="text" value="my_company.com"/> *	Authentication Realm: a name which is associated with the protected area

6. (Optional) Click **Show Advanced Fields** and change default settings as needed.

Note that the use of AES encryption is subject to import control restrictions. The version of JCE policy files bundled in the J2EE 1.4.x and later environment allows for “strong” but limited cryptography. To use the strongest AES encryption, download and install the JCE “Unlimited

Strength Jurisdiction Policy Files” from <http://java.sun.com/products/jce/javase.html#UnlimitedDownload>.

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
- http://www.windowsecurity.com/articles/Passwords_Improve_Windows_Security_Part1.html

7. Click **Next**.
8. On the Adapter Actions screen, click **Generate properties**.
9. Copy the resulting properties on the Adapter Actions screen and make them available to the application.

Refer to the *User Guide* for either the Java or .NET Integration Kits for details.
10. Click **Next**.
11. (Optional) On the Extended Adapter Contract screen, configure additional attributes as needed for this adapter instance (see “[Extending an Adapter Contract](#)” on page 203).
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.

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://www.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. These templates are discussed in the [“System Administration”](#) chapter (see [“Using Velocity Templates”](#) on page 110).

IdP Endpoints

The following sections describe the two IdP endpoints, including the query string parameters that each accepts or requires.



Note: These endpoints accept either the HTTP `GET` or `POST` methods.

/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 [“SSO Directory Service”](#) on page 303.

The following table shows the HTTP parameters for this endpoint.

PartnerSpId or PARTNER	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)
TargetResource or TARGET (optional)	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. For SAML 1.x, the value is sent to the SP as a query parameter or form control named TARGET.
InErrorResource (optional)	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 “Using Velocity Templates” on page 110).
Binding (optional)	Indicates the binding to be used; allowed values are URIs defined in the SAML specifications.
IdpAdapterId (optional)	Allows an application to call out what IdP adapter to use for authentication (in a configuration with multiple IdP adapters).
RequestedFormat (optional - SAML 2.0)	Allows limited control over the NameId format.

/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.

TargetResource (optional)	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 IdP Events screen.
------------------------------	---

InErrorResource (optional)	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 “Using Velocity Templates” on page 110).
-------------------------------	--

/idp/writecdc.ping

This endpoint is used in the SAML 2.0 IdP Discovery functionality. 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.

The following table shows the HTTP query parameter for this endpoint.

TargetResource (optional)	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.
------------------------------	--

SP Endpoints

The following sections describe the three SP endpoints, including the query string parameters that each accepts or requires.



Note: These endpoints accept either the HTTP GET or POST methods.

/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 [“SSO Directory Service”](#) on page 303.

The following table shows the HTTP parameters for this endpoint.

PartnerIdpId (required if more than one IdP connection is configured)	The federation ID of the IdP that will authenticate the user and issue an assertion.
--	--

TargetResource or TARGET (optional)	This parameter indicates where the end-user is redirected after a successful SSO. If this parameter is not included in the request, PingFederate uses as a default the URL for a successful SSO as entered on the SP Events screen.
InErrorResource (optional)	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 “Using Velocity Templates” on page 110).
SpSessionAuthn AdapterId (optional)	The explicit SP adapter instance ID indicating the adapter to use to create an authenticated session or security context.
ForceAuthn (optional - SAML 2.0)	This parameter controls the attribute of the same name in the AuthnRequest. (The default is false.)
IsPassive (optional - SAML 2.0)	This parameter controls the attribute of the same name in the AuthnRequest. (The default is false.)
AllowCreate (optional - SAML 2.0)	Controls the value of the AllowCreate attribute of the NameIDPolicy element in the AuthnRequest. (The default is true.)
RequestedFormat (optional - SAML 2.0)	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.
RequestedACSIdx (optional - SAML 2.0)	The index number of your Assertion Consumer Service where you want the assertion to be sent.
RequestedBinding (optional - SAML 2.0)	Indicates the binding requested for the response containing assertion; allowed values are URIs defined in the SAML specifications.

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 SP Adapter Instances”](#) on page 205).

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.

TargetResource (optional)	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 Events screen.
InErrorResource (optional)	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 “Using Velocity Templates” on page 110).
SpSessionAuthnAdapterId (optional)	The SP adapter instance ID indicating which session to terminate and which IdP will receive the logout request.
SourceResource (optional)	A URL indicating the origin of the logout request. It is mapped to an adapter ID in order to designate which session to terminate.

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 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 it 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”](#) on page 32). This endpoint is similar to `/sp/startSSO.ping` and accepts the same parameters, with the exception of `PartnerIdpId` (see [“/sp/startSSO.ping”](#) on page 291). 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.

The following table shows the HTTP Parameters for this endpoint.

SubjectDN	From the user's x.509 certificate, the Subject DN uniquely identifies the user to the IdP. The parameter should be URL encoded.
IssuerDN (optional)	From the user's x.509 certificate, the IssuerDN uniquely identifies the entity that issued the user's certificate. The parameter should be URL encoded.
PartnerIdpId (optional)	Used to identify the specific IdP partner to which the Attribute Query should be sent. If this parameter is not present, the SubjectDN and IssuerDN is used to determine the proper IdP.
Appld	The unique identifier of the initiating application.
SharedSecret	Used to authenticate the initiating application. The Appld and SharedSecret must both match the application authentication settings within the PingFederate server.
RequestedAttrName (optional)	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.

Clustering and Failover Deployment

PingFederate provides support for server-clustering deployment to facilitate high availability of identity federation services and/or hot failover, which permits PingFederate to continue processing transactions in the event of a hardware or software failure. Clustering may also be used to support seamless hardware upgrades or software patches to a PingFederate server without an interruption in service.



Note: For best performance, Ping Identity recommends that multicast transmissions occur in a dedicated network environment.

In addition to straightforward, “out-of-the-box” deployment and server configuration, PingFederate provides SDK extensibility for special situations.

This appendix covers:

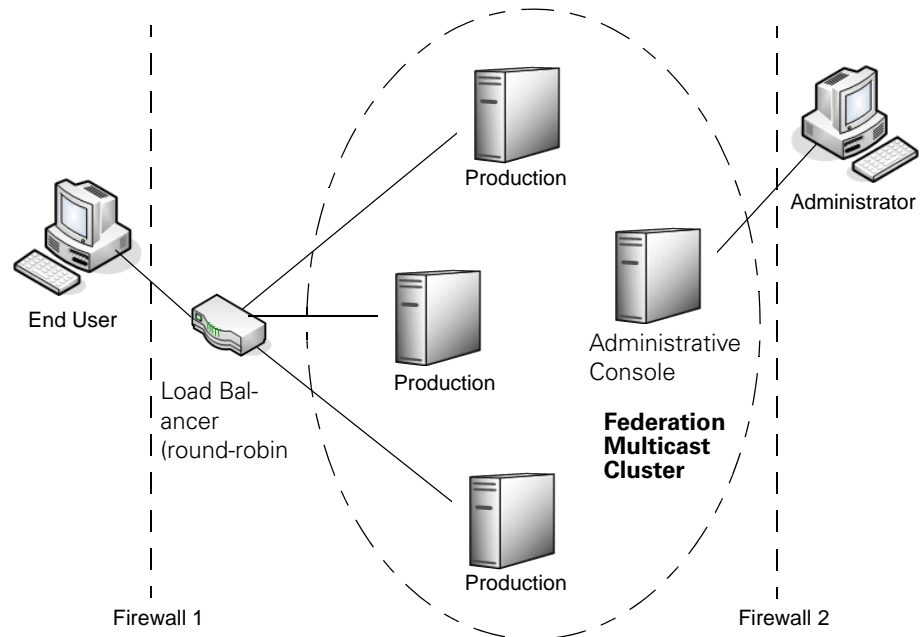
- [Deployment Scenarios](#)
- [Server Installation](#)
- [SDK Clustering Extensibility](#)

Deployment Scenarios

PingFederate’s clustering capabilities support innumerable deployment scenarios, depending on your network environment and requirements. This section provides three examples:

- [“Basic Clustering”](#) on page 296
- [“Subclustering”](#) on page 297
- [“Configuration Deployment”](#) on page 300

Basic Clustering

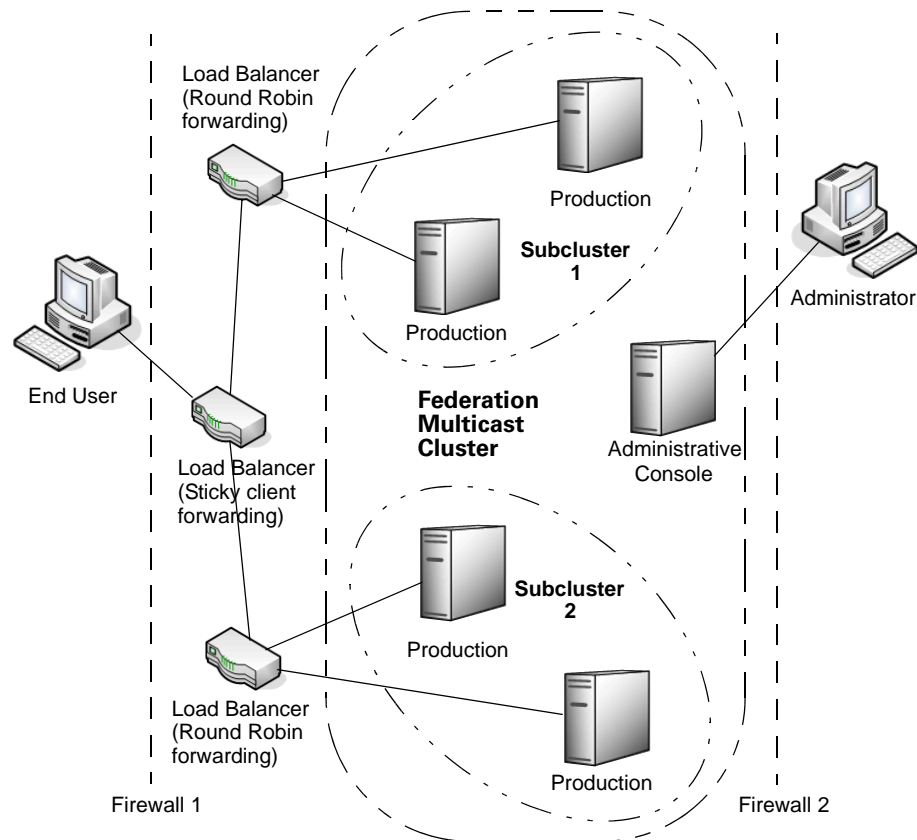


In this scenario four PingFederate servers and a load balancer are located in a data center. The load balancer (not included with PingFederate) is the only URL available to the Internet—consult your load-balancer documentation for specific configuration instructions.

Server settings and connection configuration for the cluster are done inside the corporate network using the administrative console server. The console server is accessible only in the data center or from inside the corporate network. The configuration on the console is pushed out to the production servers via multicast (see [“Configuration Deployment”](#) on page 300).

Any of the three production servers is capable of servicing any transaction because all necessary state data is shared among them via multicast. If a server fails for any reason, the other two can keep processing.

Subclustering



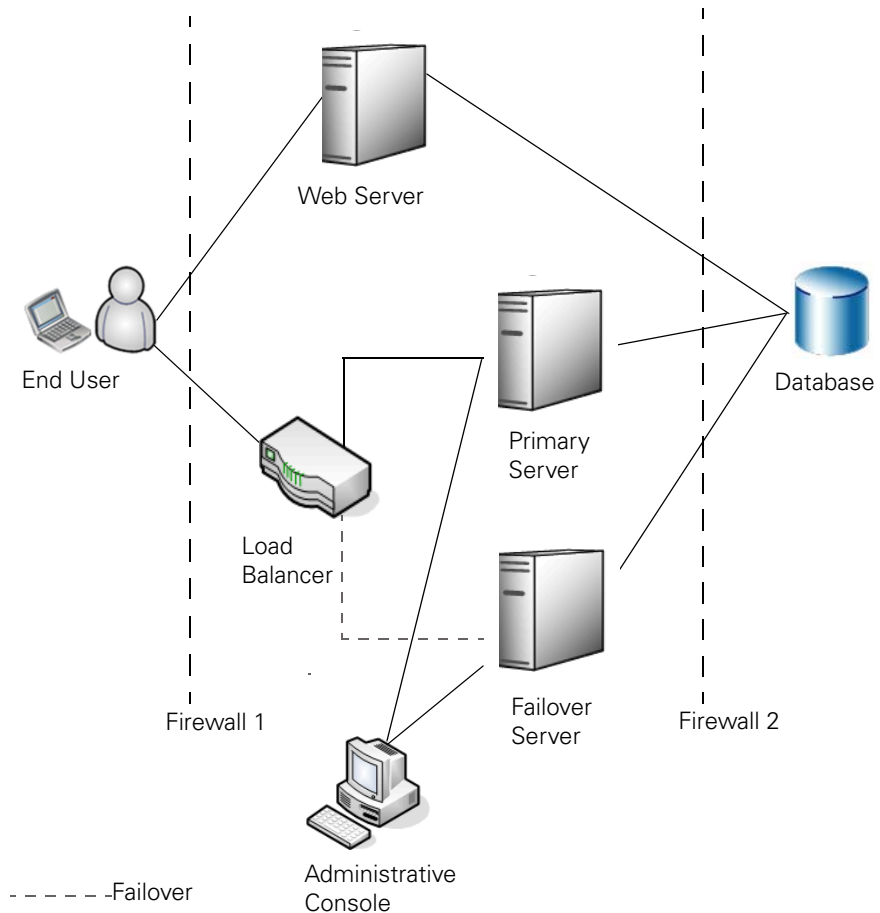
In this scenario there are five servers. The production cluster is divided into two subclusters, each using separate multicast IPs within their own subcluster, though sharing a common multicast IP for the cluster as a whole (see [“Server Installation”](#) on page 299). This effectively partitions the full cluster into logical groups. Runtime state is replicated within a sub-cluster but only a subset of the runtime state is replicated across the full cluster.

This subcluster partitioning allows for greater horizontal scalability of the cluster while maintaining an acceptable level of reliability. The specific configuration of the load balancer(s) in front of the cluster makes this possible. HTTP client requests must always be “sticky”-routed to the same subcluster group for this configuration to work properly.

The diagram above depicts three physical load balancers (for illustration only: one physical machine might be sufficient). The first is exposed to the Internet and routes client requests to one of the other two balancers using any form of sticky-client technique (the same client is always routed to the same subcluster). The second layer of load-balancers sends round-robin transactions to the production servers in their respective subcluster. (Load balancers are not part of the PingFederate distribution. Consult your load-balancer documentation for specific information.)

Dedicated Failover

The recommended PingFederate clustering deployment to handle purely failover behavior consists of three servers running PingFederate in a modified cluster configuration: an administrative console server, a primary server, and a failover server.



As in basic clustering, the administrative console is configured to run only the administrative user interface and is used to send configuration information to the other PingFederate servers in the cluster.

The difference between basic clustering deployment and failover-only deployment is that only one server processes transactions, as controlled by the load balancer (not part of the PingFederate product). The failover server stands ready to assume processing if the primary server fails.



Note: The administrative console can be configured to run on the primary server. This is not recommended, however, due to potential synchronization lapses.

For failover deployment, configure the load balancer, whether a software or hardware solution, to route all transactions to the primary PingFederate server. (Consult your load-balancer documentation for specific configuration

instructions.) If an incident prevents the server configured as the primary PingFederate server from communicating with the load balancer, traffic is automatically routed to the failover machine.

Server Installation

For each machine in a cluster, the first step is to follow the PingFederate installation procedures (see [“Installing PingFederate”](#) on page 54).



Note: Install one license on any machine in the cluster. The key will be pushed out to the other servers (see [“Configuration Deployment”](#) on page 300).

Once the installation of the servers is complete, each instance must be configured to work in a cluster:

To configure the PingFederate server to run only the administrative console:

- ▶ Edit the following properties in the `run.properties` file located in the `<PF_install_dir>/pingfederate/bin` directory on the administrative console server:
 - Set `pf.operational.mode` to `CLUSTERED_CONSOLE`.
 - Set `pf.cluster.multicast.ip` to the multicast IP address of the PingFederate servers in the cluster.

To configure production server instances of PingFederate:

- ▶ In the `run.properties` file located in the `<PF_install_dir>/pingfederate/bin` directory on each production server:
 - Set `pf.operational.mode` to `CLUSTERED_ENGINE`.
 - Set `pf.cluster.multicast.ip` to the multicast IP address of all the PingFederate servers in the cluster.
 - If you are *not* using subclustering features, set `pf.subcluster.multicast.ip` to the same multicast IP address used above, or delete the property.
 - If you *are* using subclusters, set `pf.subcluster.multicast.ip` to the multicast ID of the subcluster for this server (see [“Subclustering”](#) on page 297).



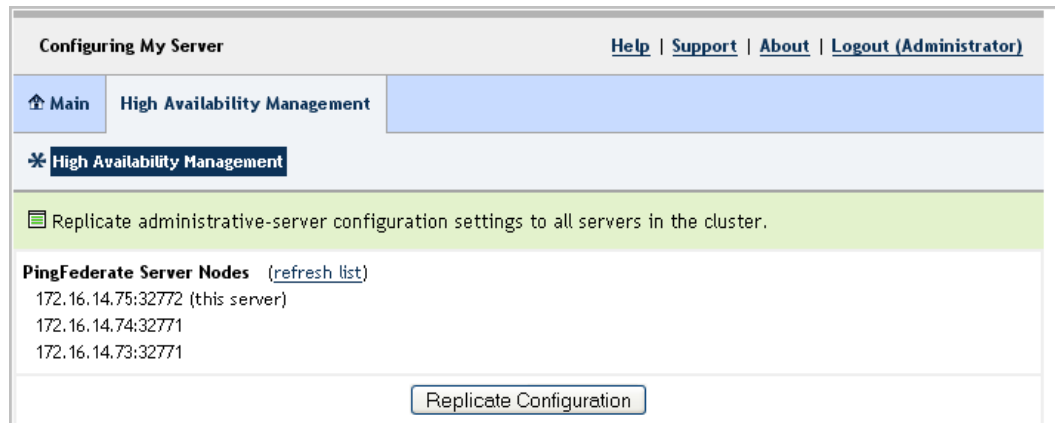
Note: The range of reserved IP addresses for `pf.cluster.multicast.ip` and `pf.subcluster.multicast.ip` is 224.0.0.0 to 239.255.255.255.

Start all PingFederate servers (see [“Starting and Stopping PingFederate”](#) on page 90).

At this point, none of the servers has enough configuration information to operate in production. The next step, “[Configuration Deployment](#)”, makes all the PingFederate servers in the cluster operational.

Configuration Deployment

Connection parameters, data store connections, and other system configuration settings must be deployed to all PingFederate servers in a cluster. Configuration is handled from the administration console server, where you can distribute configuration information and updates to the other PingFederate servers.



When you have configured or reconfigured PingFederate on the administrative console server:

1. Click **High Availability Management** on the Main Menu screen.
2. Click **Replicate Configuration** to push the configuration settings to each server in the cluster.

The servers will immediately receive these settings and update their local configurations.



Important: You must push out any changes in your configuration. Changes are not replicated automatically, except in the case where a new production server is added, in which case the most recently pushed out configuration will be added.

SDK Clustering Extensibility

The PingFederate SDK allows developers to write their own implementations of any of the following services associated with federation message processing in a clustered-server environment:

- Account linking
- Pseudonym management
- IdP session registry
- SP session registry
- POST assertion replay prevention
- Artifact message storage
- Inter-request state management

See the `README.txt` file in the `/sdk` directory for more information.

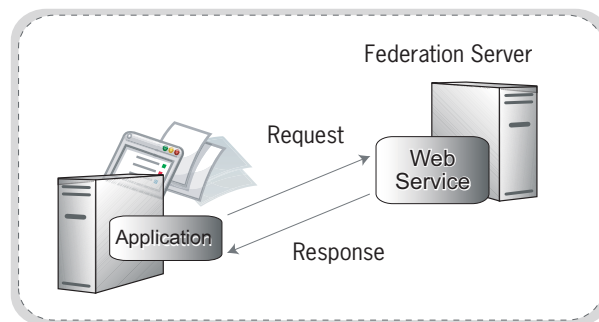
SSO Directory Service

PingFederate Single Sign-on (SSO) Directory Service is a Web service that uses the Simple Object Access Protocol (SOAP) to allow applications to retrieve configuration data from the PingFederate server. This allows applications to avoid storing and maintaining the data locally. Three types of data can be retrieved:

- A list of IdP partners
- A list of [adapter](#) instances
- A list of SP partners

The SSO Directory Service provides information required to integrate 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 figure below illustrates the SSO Directory Service interface:



The WAR file for this module, `pf-ws.war`, is located in the `pingfederate/server/default/deploy` directory. This file is deployed by default. If you do not want to use the directory service, remove the file.

The service endpoint is `pf-ws/services/SSODirectoryService`.



Note: You can require that applications accessing this endpoint first authenticate themselves, via HTTP Basic, to the PingFederate server (see [“Application Authentication”](#) on page 120).

The Web Services Description Language (WSDL) document describing this service can be retrieved from:

`http(s)://<your-pingfederate-server>:<port number>/pf-ws/services/SSODirectoryService?wsdl`

You can retrieve the three lists using the following methods:

- `getIDPList` – Returns a list of active IdPs containing each IdP’s entity ID and company name
- `getAdapterInstanceList` – Returns a list of adapter instances containing an ID and name
- `getSPList` – Returns a list of active SPs containing each SP’s entity ID and company name



Note: These three methods do not require input parameters.

SOAP Request and Response Example

Your application must send a SOAP request to the PingFederate server specifying which list you need for your integration. The following is an example of a typical SOAP request for an IdP list:

```
<?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="http://localhost:9030/ssodir/services/
SSODirectoryService"/>
  </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 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>
    <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>

```

Code Example

When you integrate an application with PingFederate, you can use the SSO Directory Service to generate an IdP list, an adapter instance list, or an SP list. The code needed to create these three lists is similar.

The following is an example of Java code that 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 Java code and the Apache Axis library.

This example includes optional code for authentication to the PingFederate server (see [“Application Authentication”](#) on page 120). We recommend that you use HTTPS when including credentials.

```
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(
            "http://localhost:9030/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);
        }
    }
}
```

Using the SafeNet Luna HSM

Federal Information Processing Standard (FIPS) 140-2 requires the storage and processing of all keys and certificates on a certified cryptographic module. To meet this requirement, PingFederate is engineered and tested with the standard-compliant SafeNet Luna SA Hardware Security Module (HSM).

If your site requires FIPS 140-2 compliance, before running PingFederate you must install and configure the Luna SA HSM according to the manufacturer's documentation. Once this process is complete, follow the steps outlined below to configure PingFederate to interact with the HSM for key generation, storage, and operation.

Note that some restrictions exist in the operation of PingFederate when using an HSM:

- Private encryption keys are not exportable. When configured for use with the HSM, administrative-console options for this feature are disabled. Only the public portion of generated keys is exportable.
- The XML encryption feature is not available (see [“XML Encryption”](#) on page 46).
- Not all cipher suites in a standard Java configuration are available. They are limited to those listed in the file named `com.pingidentity.crypto.LunaJCEManager.xml` located in the `<PF_install_dir>/server/default/data/config-store` directory.
- When using the Configuration Archive feature, any keys, certificates, or objects generated and stored on the HSM prior to saving a configuration archive must continue to exist unaltered when the archive is restored (see [“Using the Configuration Archive”](#) on page 99). In other words, any deletion or creation of objects on the HSM not executed via the PingFederate user interface will not be recognized or operational.

For example, during the course of normal PF operation you create and save objects A, B and C to the HSM and create a data archive that contains

references to those objects. If you then delete object C and attempt to recover it via the data archive, PingFederate will fail, producing various exceptions. Because the data archive contains a reference to the object and the object has been deleted from the HSM, it is not possible to use that data archive again.

To use PingFederate with the Luna SA HSM:

1. Install and configure your SafeNet Luna SA HSM, including the optional package for Java (referred to as the JSP), according to SafeNet's instructions.

This includes the creation of a partition, creation of a Network Trust Link (NTL), and assignment of a client to a partition. Ensure that you can perform the `vtl verify` command indicating that you are communicating securely and properly to the HSM.

Delete any unnecessary keys or objects that may have been created while testing communication to the HSM from the host that will run PingFederate.

Note the password that was used to open communication to the HSM via the NTL. You will need this for your installation of PingFederate.

2. To enable the Java interface, copy the following files to your Java installation:

For Windows:

Copy these files from the `Program Files\LunaSA\JSP\lib` folder into your `JAVA_HOME\jre\lib\ext` folder.

- `LunaAPI.dll`
- `LunaJCASP.jar`
- `LunaJCESP.jar`

For UNIX/Linux:

Copy these files from the `/usr/lunasa/jsp/lib` directory into your `JAVA_HOME/jre/lib/ext` directory.

- `libLunaAPI.so`
- `LunaJCASP.jar`
- `LunaJCESP.jar`

SafeNet provides some sample Java applications that can optionally be run to ensure that the Java/HSM interface is working properly prior to installing PingFederate. Please contact SafeNet Support for more information.

3. Install PingFederate on the network interconnected to the HSM (see [“Installing PingFederate”](#) on page 54).
4. In the `<PF_install_dir>/server/default/data` directory, delete files with the extension `.jks`, specifically:
 - `ping-dsig.jks`
 - `ping-ssl-server.jks`
 - `ping-ssl.jks`
 - `ping-trust.jks`

-
5. In the `hivemodule.xml` file in the `<PF_install_dir>/server/default/conf/META-INF` directory, comment out this section of XML code (comment indicators are in **bold**):

```
<!-- Use this service-point if you are using default certificate
storage -->
<!--
    <service-point id="JCERManager"
interface="com.pingidentity.crypto.JCERManager">
        <invoke-factory>
            <construct
class="com.pingidentity.crypto.SunJCERManager"/>
        </invoke-factory>
    </service-point>
-->
```

6. Just below the code in the last step, uncomment this code:

```
<service-point id="JCERManager"
interface="com.pingidentity.crypto.JCERManager">
    <invoke-factory>
        <construct
class="com.pingidentity.crypto.LunaJCERManager"/>
    </invoke-factory>
</service-point>
```

7. Later in the same file, comment out this section:

```
<!-- Use this service-point if you are using default certificate
storage -->
<!--
    <service-point id="CertificateService"
interface="com.pingidentity.crypto.CertificateService">
        <invoke-factory>
            <construct
class="com.pingidentity.crypto.CertificateServiceImpl"/>
        </invoke-factory>
    </service-point>
-->
```

8. Just below the code in the last step, uncomment this code:

```
<service-point id="CertificateService"
interface="com.pingidentity.crypto.CertificateService">
    <invoke-factory>
        <construct
class="com.pingidentity.crypto.LunaCertificateServiceImpl"/>
    </invoke-factory>
</service-point>
```

9. Save and close the `hivemodule.xml` file.

10. In the `run.properties` file found in the `<PF_install_dir>/bin` directory, change the value of the `pf.hsm.mode` property at the bottom of this file from OFF to LUNA, as shown below:

```
#
```

```
# This property denotes FIPS mode. Current values are:
# LUNA - denotes a SafeNet implementation
# OFF - Use the default Sun keystore/JCE implementation
#
pf.hsm.mode=LUNA
```

11. Save and close the `run.properties` file.

12. From the `<PF_install_dir>/bin` directory, run the `hsmpass.bat` batch file for Windows or the `hsmpass.sh` script for UNIX/Linux.

Enter the NTL password when prompted (see [Step 1](#)).

This procedure sets and securely stores the password for NTL communication to the HSM from PingFederate.

Be aware that Java echoes each key press on the console; others might be able to see the password.

13. In your Java SDK directory, open the file `java.security` in the `jre/lib/security` directory and add the two lines in boldface to *top of the list* of security providers:

```
# List of providers and their preference orders (see above):
security.provider.1=com.chrysalisits.crypto.LunaJCAProvider
security.provider.2=com.chrysalisits.cryptox.LunaJCEProvider
```

Renumber the existing security providers in the list accordingly. For example, a correctly configured list would look similar to this:

```
security.provider.1=com.chrysalisits.crypto.LunaJCAProvider
security.provider.2=com.chrysalisits.cryptox.LunaJCEProvider
security.provider.3=sun.security.provider.Sun
security.provider.4=sun.security.rsa.SunRsaSign
security.provider.5=com.sun.net.ssl.internal.ssl.Provider
security.provider.6=com.sun.crypto.provider.SunJCE
security.provider.7=sun.security.jgss.SunProvider
security.provider.8=com.sun.security.sasl.Provider
```

14. Save and close the `java.security` file.

This completes the steps required to configure PingFederate for use with the Luna SA. You may start the PingFederate server in the normal way and proceed as you would for any other installation (see [“Running PingFederate for the First Time”](#) on page 54).

Troubleshooting

Basic troubleshooting tips are provided here for you to overcome the most common difficulties experienced by both administrators and users. Please contact Ping Identity at 303.468.2882 or info@pingidentity.com for further information.

This appendix contains the following sections:

- [“Data Stores”](#) on page 312
- [“Installation”](#) on page 313
- [“Protocol”](#) on page 313
- [“Server”](#) on page 313

Data Stores

Table 8: Troubleshooting Data Stores

Problem	Solution
When setting up the JDBC data store, a connection cannot be established.	Verify that the proper drivers and connectors have been installed. Also, verify the connection URL, username, and password. If unsuccessful, contact your database administrator.
Cannot connect to the JDBC data store when using Netscape.	When a user sets Netscape to remember logon information, the Username and Password are automatically inserted into the JDBC Username and Password fields. When prompted to remember passwords, select No. If you have inadvertently selected Yes, you must remove the password in the Password Manager. To remove the password in Netscape, select Tools > Password Manager > Manage Stored Passwords and delete the entry.
Cannot connect to a Directory Service with the LDAP protocol.	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 <code>JAVA_HOME/jre/lib/security/cacerts</code> keystore (consult your Java documentation and the Java keytool documentation). The trusted certificates that are stored and accessed through the PingFederate console (My System Settings > Trusted Certificates) are used only with SOAP connections used for the Artifact Profile and are not used for other SSL/TLS connections. Verify the connection URL, port, principal, and credentials. If unsuccessful, contact your system administrator.

Installation

Table 9: Troubleshooting Installation

Problem	Solution
Error message "Not enough memory on the server"	Verify that there is at least 512 MB of RAM installed on the server (see "System Requirements" on page 52).
Exception in thread "main" java.lang.NoClassDefFo undError.	Make sure your instance of PingFederate is installed in a directory structure that does not contain spaces.

Protocol

Table 10: Troubleshooting Protocol

Problem	Solution
Certificates unexpectedly expire.	Verify that the server clocks are synchronized on both sides of the federation.

Server

Table 11: Troubleshooting Server

Problem	Solution
PingFederate does not start.	Make sure that Java is installed.

Glossary

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 of 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: a Standard 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.

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.

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.

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 standards-based means of providing authentication information and other user or system attributes across Internet security domains.

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

A Java API that allows Java programs to interact with databases.

keysize

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.

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

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.

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

A standard XML-based framework for the exchange of authentication and authorization information across security 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.

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. For example, the user is logged on to the SP and requests protected resources on an SP. The SP starts the SSO process.

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 implement session persistence.

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 a 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

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.

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

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.

List of Acronyms

ACS	Assertion Consumer Service	LDAP	Lightweight Directory Access Protocol
API	Application Programmer Interface	O	Organization
ARS	Artifact Resolution Service	OASIS	Organization for the Advancement of Structured Information Standards
CA	Certificate Authority	OU	Organizational Unit
CSR	Certificate Signing Request	PKI	Public Key Infrastructure
DBMS	Database Management System	RDBMS	Relational Database Management System
DMZ	Demilitarized Zone	RST	<RequestSecurityToken> WS-Federation XML element
DN	Distinguished Name (certificate identifier)	SAML	Security Assertion Markup Language
DNS	Domain Name System	SDK	Software Development Kit
EIM	Enterprise Identity Management	SP	Service Provider
GUI	Graphical User Interface	SLO	Single Logout
HTTP	HyperText Transfer Protocol	SOA	service-oriented architecture
HTTPS	Secure HyperText Transfer Protocol	SOAP	Simple Object Access Protocol
IdM	Identity Management	SQL	Structured Query Language
IdP	Identity Provider	SSL	Secure Sockets Layer
IP	Internet Protocol	SSL/TLS	Secure Sockets Layer/Transport Level Security
J2SDK	Java 2 Software Development Kit		
JDBC	Java Database Connectivity		

List of Acronyms

SSO	Single Sign-On
SSTC	Security Services Technical Committee (of OASIS)
TCP	Transmission Control Protocol
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
XML	Extensible Markup Language

Index

A

- access control [101](#)
- account linking [37](#)
 - configuring [225](#)
 - SAML [34](#)
- account mapping [37](#)
- accounts, administrator [101](#)
- activation
 - for IdP connection [264](#)
 - for SP connection [192](#)
- adapters
 - commercial [40](#)
 - configuring
 - IdP [124](#)
 - SP [198](#)
 - integration [39](#)
 - mapping
 - configuring [241](#)
 - defining [228](#)
 - URLs to SP [205](#)
 - selecting for mapping [229](#)
 - selecting SP data stores for [230](#)
 - standard [39](#)
- administrative console
 - initializing [54](#)
 - navigating [65](#)
- administrators, adding [102](#)
- affiliations, SP [193](#)
- application endpoints
 - IdP [131](#)
 - SP [207](#)
- architecture
 - enterprise [9](#)
 - failover [298](#)

- archiving data [99](#)
- artifact
 - artifact/artifact, SP-initiated SSO [27](#)
 - artifact/POST, SP-initiated SSO [23](#)
 - IdP-initiated SSO [18](#), [30](#)
 - lifetime, setting for IdP connections [248](#)
 - lifetime, setting for SP connections [176](#)
 - POST/artifact, SP-initiated SSO [25](#)
 - redirect/artifact, SP-initiated SSO [26](#)
 - use [15](#)
- Assertion Consumer Service URLs [171](#)
- assertions
 - about [14](#)
 - content [147](#)
 - lifetime [141](#)
 - mapping attributes to [153](#)
- attribute authority (XASP), configuring [253](#)
- attribute contract [41](#)
 - IdP configuration [169](#)
 - SP configuration [226](#)
 - to SP [149](#)
- attribute query
 - configuring for IdP connection [253](#)
 - configuring for SP connection [180](#)
 - mapping names [254](#)
- attribute sources
 - custom [84](#)
 - to IdP [154](#)
- attributes
 - contract
 - defining [95](#)
 - mapping
 - about [37](#)
 - for SP connection [165](#), [182](#)
 - using expression for [169](#), [241](#)
 - masking in log files [31](#), [42](#), [91](#)

- overview [40](#)
- audit logging [48](#)
- authentication
 - digital signatures [46](#)
 - HTTP Basic [46](#)
 - SOAP [46](#)

B

- back-channel settings
 - for IdP connections [256](#)
 - for SP connections [185](#)
- backups, configuration [99](#)
- bindings
 - allowable
 - for IdP connection [248](#)
 - for SP connection [175](#)
 - artifact [15](#)
 - configuring
 - for IdP connection [220](#)
 - for SP connection [141](#)
 - POST [15](#)
 - redirect [15](#)
 - SOAP for IdP connection [256](#)
 - SOAP for SP connection [185](#)
- browser requirements [52](#)
- buttons, administrative console [67](#)

C

- certificate authority (CA) [111](#)
- certificates
 - authority, verifying [46](#)
 - considerations [47](#)
 - digital signing [118](#)
 - for metadata [96](#)
 - for SOAP authentication [47](#)
 - management [43](#)
 - selecting XML decryption (SP-to-IdP) [263](#)
 - selecting XML encryption (SP-to-IdP) [262](#)
 - SSL [113](#)
 - SSL client [115](#)
 - verifying [46](#)
- checklist, for federation [46](#)
- common domains [87](#)
- configuration
 - archive [99](#)
 - failover [300](#)
 - mirroring [295](#)
- configuration screens
 - navigating [65](#)
- connection list
 - for IdP connection [213](#)
 - for SP connection [134](#)
- connections
 - SAML 2.0

- IdP connection steps [215](#)
- SP connection steps [137](#)
- console buttons [67](#)
- consumer service URL [49](#)
- cookies
 - common domain [87](#)
- credentials
 - saving configuration
 - for IdP connection [264](#)
 - for SP connection [192](#)
- custom data stores
 - adapter actions [85](#)
 - filters (IdP connections) [240](#)
 - filters (SP connections) [165](#)
 - selecting fields (IdP connections) [240](#)
 - selecting fields (SP connections) [165](#)

D

- data exchange
 - automated [49](#)
 - for IdP connection [49](#)
 - for SP connection [49](#)
- data migration [99](#)
- data store [48](#)
 - compatibility [52](#)
 - description [42](#)
 - introduction [77](#)
 - JDBC configuration [79](#)
 - LDAP configuration [82](#)
 - selecting
 - for IdP connection [232](#)
 - for SP connection [155](#)
 - setup for IdP connection [231](#)
 - troubleshooting [312](#)
- data stores
 - for attribute query profile [182](#)
- database
 - filter
 - for IdP connection [234](#)
 - for SP connection [159](#)
 - for IdP connection [233](#)
 - for SP connection [157](#)
- decryption keys
 - for IdP connections [263](#)
 - for SP connections [191](#)
- default target URL [172](#)
- defederation [38](#)
- deployment [47](#)
 - diagrams [56](#)
- digital signature
 - about [43](#)
 - keys and certificates [118](#)
 - policy [44](#)
- digital signatures
 - authentication method [46](#)
- digital signing

- for IdP connection [259](#)
- for metadata [96](#)
- for SP connection [188](#)
- directory search
 - for IdP connection [236](#)
- directory service
 - code example [305](#)
 - SOAP example [304](#)
 - SSO [303](#)
- discovery, IdP [86](#)

E

- email
 - addresses [72](#)
 - options [71](#)
 - server configuration [100](#)
- enable
 - IdP functionality [69](#)
 - SP functionality [69](#)
- encryption, XML [178](#), [251](#)
- endpoints
 - IdP application [207](#)
 - protocol, IdP [131](#)
 - protocol, SP [210](#)
- endpoints, IdP application [131](#)
- entity ID [75](#)
- error handling, at runtime [110](#)
- error message, SLO [130](#)
- expressions, using for attribute mapping [169](#), [241](#)

F

- failover [298](#)
 - configuration [300](#)
 - deployment [295](#)
 - installation [299](#)
- failsafe attribute source [168](#)
- federated identity [5](#)
- federation
 - checklist [46](#)
 - defederation [38](#)
 - ID [49](#)
 - identity [5](#)
 - roles, choosing [74](#)
 - URLs [75](#)
- filter
 - for IdP connection [234](#)
 - for SP connection [159](#)
 - LDAP for IdP connection [238](#)

G

- general information
 - for IdP connection [217](#)

- for SP connection [139](#)

H

- hardware requirements [53](#)
- host names, virtual [105](#)
- HTTP Basic Authentication [46](#)

I

- identity federation [5](#)
- identity mapping [225](#)
 - about [37](#)
 - configuration [147](#)
- identity provider [14](#)
- IdP
 - adapters [124](#)
 - connections
 - activation [264](#)
 - introduction [197](#)
 - default URL [130](#)
 - enable [69](#)
 - LDAP adapter [281](#)
 - SLO error message [130](#)
- IdP Discovery
 - common domain service [87](#)
 - configuring [86](#)
 - local domain service [88](#)
 - selecting [74](#)
- IdP-initiated SLO [31](#)
 - configuring
 - for IdP connection [223](#)
 - for SP connection [145](#)
- IdP-initiated SSO
 - Artifact [18](#), [30](#)
 - configuring
 - for IdP connection [222](#)
 - for SP connection [144](#)
 - POST [16](#), [28](#)
- importing
 - metadata
 - for IdP connection [216](#)
 - for SP connection [138](#)
- installation
 - failover [299](#)
 - federation server [54](#)
 - J2SDK [53](#)
 - license key [106](#)
 - Linux service [58](#)
 - troubleshooting [313](#)
 - Windows service [58](#)
- integration kits [39](#)
 - introduction [8](#)

J

J2SDK [53](#)

installation [53](#)

Java requirements [53](#)

JDBC

configuration [79](#)

K

keys

for XML decryption (SP-to-IdP) [263](#)

for XML encryption (IdP-to-SP) [191](#)

keys and certificates

digital signature [118](#)

SSL [115](#)

L

LDAP

configuration [82](#)

directory search

for IdP connection [236](#)

for SP connection [161](#)

filter [163](#), [238](#)

LDAP adapter

configuration [279](#)

configuring for IdP [281](#)

configuring for SP [284](#)

legal agreements [46](#)

license key, installing [106](#)

licensing

email address to notify [72](#)

violation notification [71](#)

Linux

service [58](#)

starting the server [90](#)

stopping the server [90](#)

log files [90](#)

logging

administrator audit [91](#)

files [90](#)

modes [93](#)

transaction [92](#)

M

main menu [63](#)

for IdP connection [212](#)

for SP connection [133](#)

mapping

adapters

configuring [241](#)

defining [228](#)

URLs to SP [205](#)

attributes

configuring for SP connection [165](#), [182](#)

identity [37](#), [147](#), [225](#)

mapping attributes

default for SP connection [169](#)

masking attributes [42](#)

masking attributes in logs [31](#), [91](#)

message

signing [43](#)

validation [44](#), [47](#)

message URL http

[//www.pingidentity.com/products/integrationkits](http://www.pingidentity.com/products/integrationkits) [125](#)

metadata

exporting [94](#)

exporting XML encryption certificates [96](#)

importing

for IdP connection [216](#)

for SP connection [138](#)

signing [97](#)

use [49](#)

migrating configurations [99](#)

mutual settings [50](#)

N

name identifiers

SAML 2.0 [147](#)

WS-Federation [149](#)

navigation [63](#)

buttons [67](#)

notification [71](#)

O

OASIS [13](#)

Object-Graph Navigation Language (OGNL), editing [167](#), [243](#)

operating systems [52](#)

P

passive requestor profile (WS-Federation) [33](#)

passwords [71](#)

changing [105](#)

resetting [103](#)

planning checklist [46](#)

POST

artifact/POST, SP-initiated SSO [23](#)

IdP-initiated SSO [16](#), [28](#)

POST/artifact, SP-initiated SSO [25](#)

POST/POST, SP-initiated SSO [20](#)

redirect/POST, SP-initiated SSO [22](#)

use [15](#)

- profiles [15](#)
 - configuring
 - for IdP connection [220](#)
 - for SP connection [141](#)
 - IdP-initiated SSO
 - Artifact [18, 30](#)
 - POST [16, 28](#)
 - SAML 1.x [16](#)
 - SAML 2.0 [20](#)
 - saving configuration
 - for IdP connection [180, 253](#)
 - SP-initiated SSO [19](#)
 - Artifact/Artifact [27](#)
 - Artifact/POST [23](#)
 - POST/Artifact [25](#)
 - POST/POST [20](#)
 - Redirect/Artifact [26](#)
 - Redirect/POST [22](#)
- protocol endpoints
 - IdP [131](#)
 - SP [210](#)
- protocols
 - choosing [74](#)
 - troubleshooting [313](#)
- pseudonyms
 - unique values for [128](#)
 - using [34](#)

R

- redirect
 - redirect/artifact, SP-initiated SSO [26](#)
 - redirect/POST, SP-initiated SSO [22](#)
 - use [15](#)
- requirements [52](#)
 - browser [52](#)
 - data store [52](#)
 - hardware [53](#)
 - Java [53](#)
 - operating systems [52](#)
- resetting passwords [103](#)
- roles
 - about [13](#)
 - choosing IdP/SP [74](#)
- runtime configuration [107](#)

S

- SAML
 - account linking [34](#)
 - assertions [14](#)
 - bindings [15](#)
 - overview [13](#)
 - profiles, definition [15](#)
 - security [35](#)
 - selecting [74](#)

- SAML 1.1
 - IdP connection steps [215](#)
 - SP connection steps [137](#)
- SAML 1.x
 - profiles [16](#)
- SAML 2.0
 - connection steps [137, 215](#)
 - IdP connection steps [215](#)
 - profiles [20](#)
 - SP connection steps [137](#)
- scalability [7](#)
- SDK [40](#)
- security
 - back channel [47](#)
 - considerations [43](#)
 - SAML [35](#)
- server
 - ID [107](#)
 - troubleshooting [313](#)
- server ID [47](#)
- server settings [69](#)
- service provider [6, 14, 193](#)
- service URL
 - SP, for WS-Fed [173, 246](#)
- session clean-up [31](#)
- session integration considerations [48](#)
- session timeout [76](#)
- signing XML files [97](#)
- single logout (SLO) [31, 34](#)
 - service URLs
 - for SP connection [174](#)
 - session clean-up [31](#)
- single sign-on [20](#)
 - configuring [221](#)
- SLO [31, 34](#)
 - timeout [76](#)
- SOAP
 - authentication [46](#)
 - connection type
 - for IdP connection [49](#)
 - for SP connection [49](#)
 - responder URL [49](#)
- SP
 - adapters [198](#)
 - connections
 - activation [192](#)
 - introduction [123](#)
 - default URL [207](#)
 - enable [69](#)
 - LDAP adapter [284](#)
- SP-initiated SLO [31](#)
 - configuring
 - for IdP connection [224](#)
 - for SP connection [146](#)
- SP-initiated SSO [19](#)
 - Artifact/Artifact [27](#)
 - Artifact/POST [23](#)

- configuring
 - for IdP connection [223](#)
 - for SP connection [144](#)
- POST/Artifact [25](#)
- POST/POST [20](#)
- Redirect/Artifact [26](#)
- Redirect/POST [22](#)
- SSL [43](#)
 - about [45](#)
 - certificates [113](#)
 - keys and certificates [115](#)
- SSL Client Certificate Authentication
 - authentication
 - SSL [46](#)
- SSO directory service [303](#)
- standard adapters [39](#)
- standards
 - choosing [74](#)
 - supported [8](#)
- starting and stopping the server
 - Linux service [58](#)
- starting the server
 - Linux [90](#)
 - Windows [90](#)
- stopping the server
 - Linux [90](#)
 - Windows [90](#)
- summary
 - for IdP connection [264](#)
 - for SP connection [192](#)
- synchronization [48](#)
- system administration (single or multiple users) [70](#)
- system navigation [63](#)
- system requirements [52](#)

T

- target URL, default [172](#)
- templates, HTML [110](#)
- time synchronization [48](#)
- timeouts for SLO [76](#)
- transaction logging [48](#), [92](#)
- transport security considerations [47](#)
- troubleshooting [311](#)
 - data store [312](#)
 - installation [313](#)
 - protocols [313](#)
 - server [313](#)

U

- uninstalling [60](#)
- unique IDs [47](#)
- unique values, for pseudonym creation [128](#)
- upgrading PingFederate [99](#)

- user management
 - setting [70](#)
- users
 - account management [101](#)
 - adding [102](#)
 - deactivating [103](#)

V

- validating messages [47](#)
- verifying certificates [46](#)
- virtual host names [105](#)
- virtual IDs [47](#)
 - for IdP servers [134](#)
 - for SP servers [213](#)

W

- where clause
 - for IdP connection [234](#)
 - for SP connection [159](#)
- Windows service, installation [58](#)
- WS-Federation
 - about [32](#)
 - IdP connection steps [215](#)
 - selecting [74](#)
 - setting realm ID [75](#)
 - SP connection steps [137](#)
 - SP service URL [173](#), [246](#)

X

- X.509 certificates [43](#)
- XML encryption [43](#)
 - about [46](#)
 - exporting keys to metadata [96](#)
 - IdP-to-SP [178](#)
 - SP-to-IdP [251](#)
- XML files, signing [97](#)
