PingFederate®

Web Access Management Token Translator

Version 1.2

User Guide

Ping Identity®

PingFederate Web Access Management Token Translator *User Guide*
Version 1.2
August, 2013

Ping Identity Corporation
1001 17th Street, Suite 100
Denver, CO 80202
U.S.A.

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

Trademarks

Disclaimer

Document Lifetime

Ping Identity may occasionally update online documentation between releases of the related software. Consequently, if this PDF was not downloaded recently, it may not contain the most-up-to-date information. Please refer to documentation.pingidentity.com for the most current information.

From the Web site, you may also download and refresh this PDF if it has been updated, as indicated by a change in this date: August 28, 2013.

Contents

Introduction **4**
Intended Audience 4
System Requirements 4
ZIP Manifest 5
WAM Plug-ins **5**
Custom WAM Plug-in Installation 5
WAM Plug-in Installation for RSA 6
WAM Plug-in Installation for OAM 6
Token Translator Installation **7**
WS-Trust STS Processing **7**
Configuring the IdP Token Processor **9**
Configuring the SP Token Generator **13**
Creating a Custom Authentication Scheme for OAM 17
Using the STS Client SDK **17**
Java Sample Code 17

Introduction

The PingFederate Web Access Management (WAM) Token Translator provides a Token Processor and a Token Generator for use with the PingFederate WS-Trust Security Token Service (STS). The Token Processor allows an Identity Provider (IdP) STS to accept and validate a WAM session token from a Web Service Client (WSC) and then map user attributes into a SAML token for the WSC to send to a Web Service Provider (WSP). The Token Generator allows a Service Provider (SP) STS to issue a WAM session token for a WSP, including mapped attributes from an incoming SAML token.

Important: The Token Translator is designed to work with WAM products from multiple vendors. A WAM plug-in is required to connect the Token Translator with each third-party system. This kit ships with WAM plug-ins compatible with Oracle Access Manager (OAM) 10g and 11g, and with RSA Access Manager 6.1. A simple software development kit (SDK) is also included to create custom WAM plug-ins for other systems.

If you are creating a WAM plug-in for any third-party product other than OAM and RSA Access Manager, you must complete the tasks in the WAM plug-in SDK `README.txt` file located in the `<token_translator_install_dir>/sdk` directory.

Note: Ping Identity provides an SDK for enabling Web Service applications (Client or Provider) to interact with the PingFederate STS. The SDK is available for download on the Ping Identity Downloads page (`www.pingidentity.com/products/downloads.cfm`).

Intended Audience

This document is intended for system administrators with experience in the configuration and maintenance of the OAM Access Server or RSA Access Manager and other WAM tools, as well as developers with experience using JAVA SDKs. Please consult the WAM documentation tool if you encounter any difficulties in areas not directly associated with PingFederate or the WAM Token Translator.

System Requirements

The following software must be installed in order to implement the WAM Token Translator:

- PingFederate 6.x or higher
- WAM plug-in for the desired third-party system, built and deployed per the WAM plug-in SDK documentation
- Associated vendor-supplied libraries to support the WAM plug-in you are using.

  Fully functional WAM plug-ins for OAM and RSA are included in the WAM Token Translator package.
- Separate third-party Web Agent configured using the WAM server administrative software

ZIP Manifest

The distribution ZIP file for the WAM Token Translator contains the following:

- `ReadMeFirst.pdf` – contains links to this online documentation
- `/legal` – contains this document:
  - `Legal.pdf` – copyright and license information
- `/dist` – contains libraries needed to run the Token Translator:
  - `pf-wam-token-translator-1.2.jar` – the WAM Token Translator JAR file
  - `opentoken-adapter-2.5.1.jar` – OpenToken Adapter JAR file
- `/dist/oam` – contains Oracle Access Manager libraries needed to run the adapter:
  - `pf-oam-plugin.jar` – Pre-built OAM-compatible WAM plug-in JAR file
  - `PingCustomAuthPlugin.jar` – a Java-based PingFederate Custom Authentication Scheme
- `/dist/rsa` – contains RSA libraries needed to run the adapter:
  - `pf-rsa-plugin.jar` – Pre-built RSA-compatible WAM plug-in JAR file
  - `axm-runtime-api-6.1.4` - RSA API library
  - `jsafeFIPS-6.1.jar` – RSA API library
  - `jsafeJCEFIPS-6.1.jar` – RSA API library
- `/sdk` – contains build scripts, documents, libraries, and sample code to build a WAM plug-in:
  - `README.txt` – contains documentation on how to build a WAM plug-in
  - `/docs` – contains documentation on how to build a WAM plug-in
  - `/lib` – contains libraries and supporting files needed to build WAM plug-in
  - `/samples` – contains sample code used to build a WAM plug-in

WAM Plug-ins

This kit ships with WAM plug-ins compatible with OAM 10g and 11g, RSA Access Manager 6.1, as well as a simple SDK to create custom WAM plug-ins for other systems.

Custom WAM Plug-in Installation

This section describes how to deploy a custom WAM plug-in for both Token Processors and Token Generators.

1. If you are creating a WAM plug-in for any third-party WAM product not bundled with this kit, you must complete the tasks in the WAM plug-in SDK `README.txt` file located in the `<token_translator_install_dir>/sdk` directory.

Note: Contact the third-party vendor support department to obtain required third-party WAM API libraries for creating a WAM plug-in to interact with PingFederate.

2. If you are deploying for a third-party WAM product, copy the resultant WAM plug-in output JAR file `pf-<WAM_TYPE>-plugin.jar` from the `<token_translator_install_dir>/sdk/samples/<WAM_TYPE>` directory into the `<PF_install>/pingfederate/server/default/deploy` directory.

The WAM Token Translator requires a plug-in to connect with a specific WAM product (see the WAM plug-in SDK in the distribution package for sample code and more details on building the plug-in).

Note: The WAM plug-in SDK is designed specifically to connect the WAM Token Translator with a third-party WAM product, using an API provided by the vendor.

WAM Plug-in Installation for RSA

This section describes how to deploy the pre-built RSA-compatible WAM plug-in for both Token Processor and Token Generators.

1. The additional RSA API libraries for creating a WAM plug-in to interact with PingFederate are included in the `<token_translator_install_dir>/dist/rsa` directory.
    - `axm-runtime-api-6.1.4.jar`
    - `jsafeFIPS-6.1.jar`
    - `jsafeJCEFIPS-6.1.jar`
2. Copy the RSA API libraries from the `<token_translator_install_dir>/dist/rsa` into:

    `<PF_install>/pingfederate/server/default/deploy` directory.
3. Copy the `pf-rsa-plugin.jar` from the `<token_translator_install_dir>/dist/rsa` directory into:

    `<PF_install>/pingfederate/server/default/deploy`
4. Complete the Token Translator Installation prior to restarting the PingFederate server.

WAM Plug-in Installation for OAM

This section describes how to deploy the pre-built OAM-compatible WAM plug-in for both Token Processors and Token Generators.

1. Get the necessary OAM API library from the Oracle Identity Management Download site (`http://www.oracle.com/technetwork/middleware/downloads/oid-11g-161194.html`):

    `oamasdk-api.jar`
2. Copy the OAM API library provided by the vendor into the `<PF_install>/pingfederate/server/default/deploy` directory.

3. Copy the `pf-oam-plugin.jar` file from the `<token_translator_install_dir>/dist/` directory into:

   `<PF_install>/pingfederate/server/default/deploy`.

4. Complete the Token Translator Installation prior to restarting the PingFederate server (see next section).

Token Translator Installation

This section describes how to install the WAM Token Translator to configure the Token Processor, the Token Generator, or both.

Note:If you have already deployed version 2.5.1 (or higher) of the OpenToken Adapter, skip steps 1 through 3 in the following procedure.

1. Stop the PingFederate server if it is running.
2. Remove any existing OpenToken Adapter files (`opentoken*.jar`) from the directory:

   `<PF_install>/pingfederate/server/default/deploy`

   The adapter JAR file is `opentoken-adapter-<version>.jar`.

   Note: If the adapter Jar filename indicates version 2.1 or less, also delete the supporting library `opentoken-java-1.x.jar` from the same directory.

3. Unzip the token translator distribution file and copy `opentoken-adapter-2.5.1.jar` from the `/dist` directory to the PingFederate directory.

   `<PF_install>/pingfederate/server/default/deploy`

4. From this distribution, copy the following file to the `/server/default/deploy` directory in your PingFederate server installation:

   `pf-wam-token-translator-1.2.jar`

5. *If* you are running PingFederate 6.0 as a Windows service, *then*:

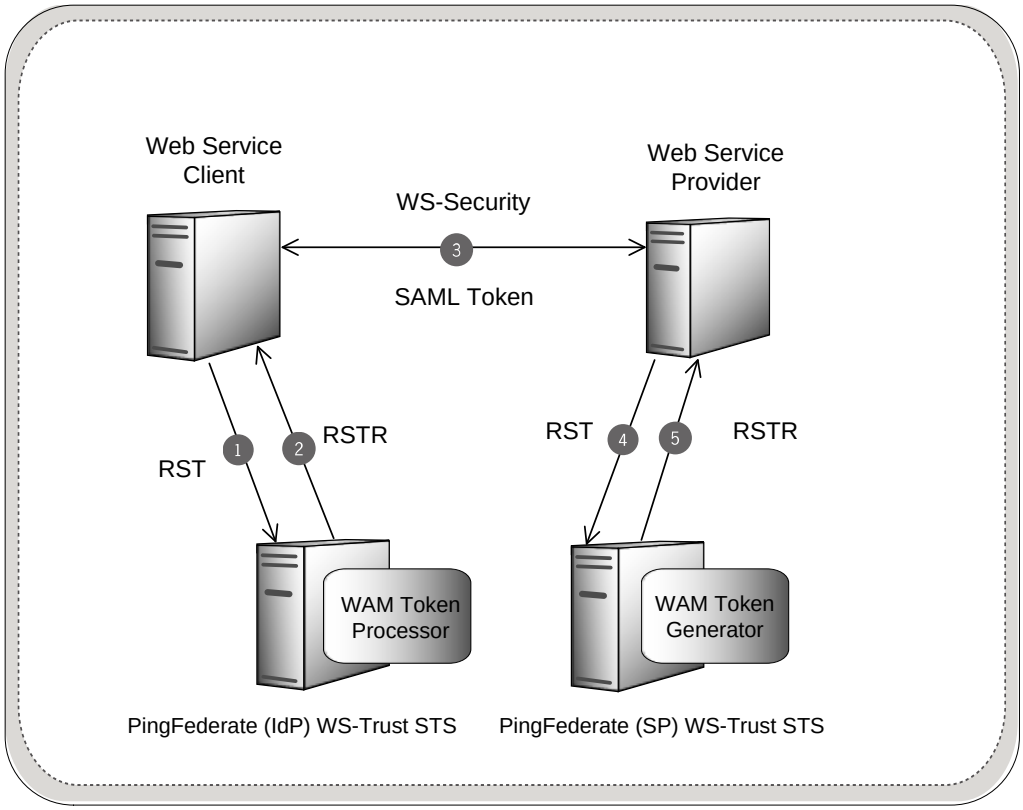
   Edit the `Java Library Path` section of the configuration file `pingfederate/sbin/wrapper/PingFederateService.conf`, adding the line:

   `wrapper.java.library.path.append_system_path=true`

6. Start the PingFederate server.

WS-Trust STS Processing

The following illustration displays a basic Web Services scenario using the PingFederate WS Trust STS in the role of both IdP and SP:



Processing Steps

1. A WSC sends a Request Security Token (RST) message containing a WAM session token to the PingFederate STS IdP endpoint.
2. The PingFederate WAM Token Processor extracts, decrypts, parses, and validates the WAM session token. If the WAM session token is valid, PingFederate maps attributes from the WAM session token into a SAML token. PingFederate issues the SAML token based on the SP connection configuration and embeds the token in a Request Security Token Response (RSTR), which is returned to the WSC.
3. The WSC binds the issued SAML token into a Web Service Security (WSS) header and sends it via a SOAP request to the WSP.
4. The WSP sends an RST Issue request containing the SAML token to the PingFederate STS SP endpoint. PingFederate validates the SAML token and, if valid, maps attributes from the SAML token into a WAM session. PingFederate issues the WAM session token based on the WAM Token Generator configuration and embeds the token in an RSTR, which is returned to the WSP.
5. The WSP receives the WAM session token in the RSTR for local domain processing.

Configuring the IdP Token Processor

If you are using PingFederate as an IdP server, configure the Token Processor using the following steps:

Important: You must first create a third-party WAM Web Agent within your WAM tool. Several properties used to configure the agent are then used on the Instance Configuration screen. Refer to your WAM documentation for details on agent configuration.

1. Log on to the PingFederate administrative console and click **Token Processors** under Application Integration Settings in the IdP Configuration section of the Main Menu.

   If you do not see **Token Processors** on the Main Menu, enable WS-Trust under Server Settings on the Roles & Protocols screen by selecting WS-Trust for the IdP role.

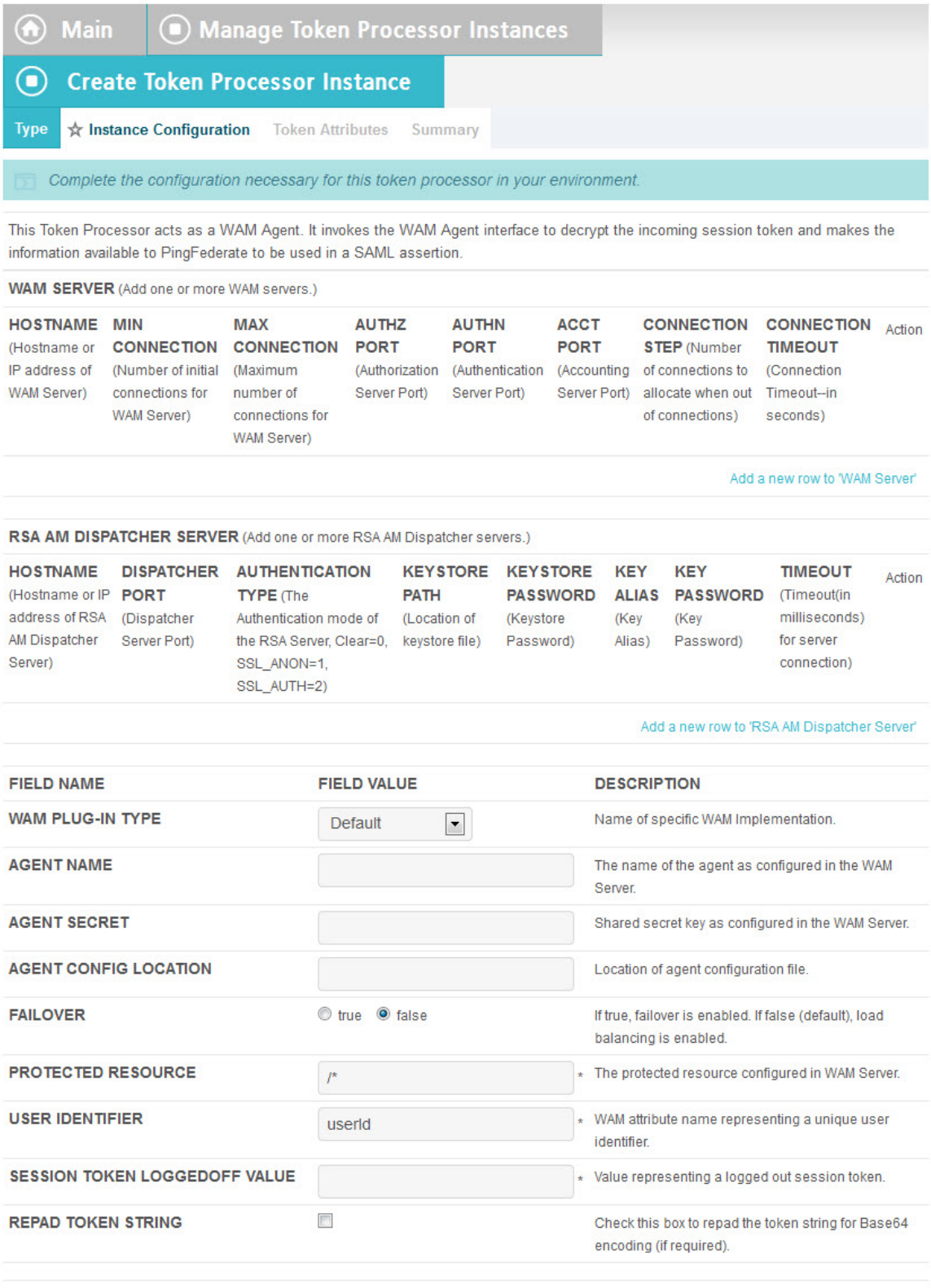
   Note: To enable token exchange, you may be prompted to provide SAML 1.x and SAML 2.0 federation identifiers for the STS on the Federation Info screen. Refer to the Federation Info screen's **Help** page for more information.

2. On the Manage Token Processor Instances screen, click **Create New Instance**.
3. On the Type screen, enter an Instance Name and Instance Id.

   The Name is any you choose for identifying this instance. The ID is used internally and may not contain spaces or non-alphanumeric characters.

4. Select WAM Token Processor 1.2 as the Type and click **Next**.

   Note: If you are configuring the adapter for a custom plug-in (not bundled with this kit), then continue to step 5. If you are configuring the RSA Dispatcher server, then continue with step 6. If you are configuring OAM, continue at step 7.


Main
Manage Token Processor Instances
Create Token Processor Instance
Type
Instance Configuration
Token Attributes
Summary
Complete the configuration necessary for this token processor in your environment.
This Token Processor acts as a WAM Agent. It invokes the WAM Agent interface to decrypt the incoming session token and makes the information available to PingFederate to be used in a SAML assertion.
WAM SERVER (Add one or more WAM servers.)
HOSTNAME (Hostname or IP address of WAM Server)
MIN CONNECTION (Number of initial connections for WAM Server)
MAX CONNECTION (Maximum number of connections for WAM Server)
AUTHZ PORT (Authorization Server Port)
AUTHN PORT (Authentication Server Port)
ACCT PORT (Accounting Server Port)
CONNECTION STEP (Number of connections to allocate when out of connections)
CONNECTION TIMEOUT (Connection Timeout--in seconds)
Action
Add a new row to 'WAM Server'
RSA AM DISPATCHER SERVER (Add one or more RSA AM Dispatcher servers.)
HOSTNAME (Hostname or IP address of RSA AM Dispatcher Server)
DISPATCHER PORT (Dispatcher Server Port)
AUTHENTICATION TYPE (The Authentication mode of the RSA Server, Clear=0, SSL_ANON=1, SSL_AUTH=2)
KEYSTORE PATH (Location of keystore file)
KEYSTORE PASSWORD (Keystore Password)
KEY ALIAS (Key Alias)
KEY PASSWORD (Key Password)
TIMEOUT (Timeout(in milliseconds) for server connection)
Action
Add a new row to 'RSA AM Dispatcher Server'
FIELD NAME
FIELD VALUE
DESCRIPTION
WAM PLUG-IN TYPE
Default
Name of specific WAM Implementation.
AGENT NAME
The name of the agent as configured in the WAM Server.
AGENT SECRET
Shared secret key as configured in the WAM Server.
AGENT CONFIG LOCATION
Location of agent configuration file.
FAILOVER
true
false
If true, failover is enabled. If false (default), load balancing is enabled.
PROTECTED RESOURCE
/*
The protected resource configured in WAM Server.
USER IDENTIFIER
userId
WAM attribute name representing a unique user identifier.
SESSION TOKEN LOGGEDOFF VALUE
Value representing a logged out session token.
REPAD TOKEN STRING
Check this box to repad the token string for Base64 encoding (if required).

5. (Only for custom plug-ins for WAM servers other than OAM or RSA) On the Instance Configuration screen, click **Add a new row to 'WAM Server'** and provide the following information into the table:
   a. Enter the Hostname or the IP address where the WAM server is running.
   b. Specify the remaining WAM server values required for your configuration.
   c. Click **Update** in the Action column.
   d. Repeat this step as needed, for additional WAM plug-ins.

   Skip the next step.

6. (Only for the RSA bundled plug-in) On the Instance Configuration screen, click **Add a new row to 'RSA AM Dispatcher Server'** and provide the following information in the table:

   Note: You must specify at least one RSA AM Dispatcher Server

   a. Enter the Hostname or the IP address and the (optional) Dispatcher Port where the RSA AM Dispatcher server is running.

      Note: You must specify the authentication method that is used by the dispatcher server. If you have specified multiple dispatcher servers, each server can have individual authentication methods.

   b. Specify the Authentication Type used by the RSA Dispatcher Server.
      - **Clear** – clear text, no encryption
      - **Anon** – anonymous SSL, SSL encryption only
      - **Auth** – mutually authenticated SSL, SSL encryption with certificate-based encryption

   c. If the selected Authentication Type is **Auth**, you must specify the following RSA server values:
      - Keystore Path – String filename of the private Keystore file (PKCS12 only)
      - Keystore Password – password for the private Keystore
      - Key Alias – the alias to your private key in the Keystore
      - Key Password – private Key Password for Keystore

   d. (Optional) Specify the Timeout value required for your configuration.
   e. Click **Update** in the Action column.
   f. Repeat this step as needed for additional RSA Servers.

7. Provide entries on the Instance Configuration screen, as described on the screen and in the following table.

Note: The selected WAM Plug-in Type may override optional/required fields. For example, if the selected WAM Plug-n Type is `OAM`, the Agent Config Location becomes a required field. Leaving this field blank generates an error message.

Field	Description
WAM Plug-in Type	Class name for the specific WAM implementation. **Note:** WAM Plug-in Type determines optional/required fields.
Agent Name	This value must match the value used when the third-party WAM Web Agent was configured.
Agent Secret	This value must match the value used when the third-party WAM Web Agent was configured.
Agent Config Location	Required for OAM, this value must contain the full path to an XML network-configuration file generated by the access-management system.
Failover	The default is false, indicating load balancing is enabled and user-session states and configuration data are shared among multiple WAM servers. Select **true** to enable failover, indicating that when one server fails, the next server is used.
Protected Resource	(Required) All files in the root directory (/*) is the default. Specify a different path to the resources in the protected realm, if necessary.
User Identifier	(Required) Defines which attribute that is parsed from the WAM session token is the user identifier for use in the assertion.
Session Token LOGGEDOFF Value	(Required) Value representing a logged-out session token.
Repad Token String	Enable this to pad the incoming session token string for Base64 encoding (if required).

8. Click **Next**.
9. (Optional) On the Token Attributes screen, select any or all attributes whose value you want to mask in the PingFederate log file.

   For more information about this screen, see Setting Pseudonym Values and Masking in the PingFederate *Administrator's Manual*. More information is available on the **Help** page.
10. Click **Next**.

11. On the Summary screen, verify that the information is correct and click **Done**.
12. On the Manage Token Processor Instances screen, click **Save**.

Configuring the SP Token Generator

If you are using PingFederate as a Service Provider (SP), configure the Token Generator using the following steps:

Important: You must first create a third-party WAM Web Agent within your WAM tool. Several properties used to configure the agent are then used on the Instance Configuration screen. Refer to your WAM documentation for details on agent configuration.

1. Log on to the PingFederate administrative console and click **Token Generators** under Application Integration Settings in the SP Configuration section of the Main Menu.

   If you do not see **Token Generators** on the Main Menu, enable WS-Trust under Server Settings on the Roles & Protocols screen by selecting WS-Trust for the SP role.

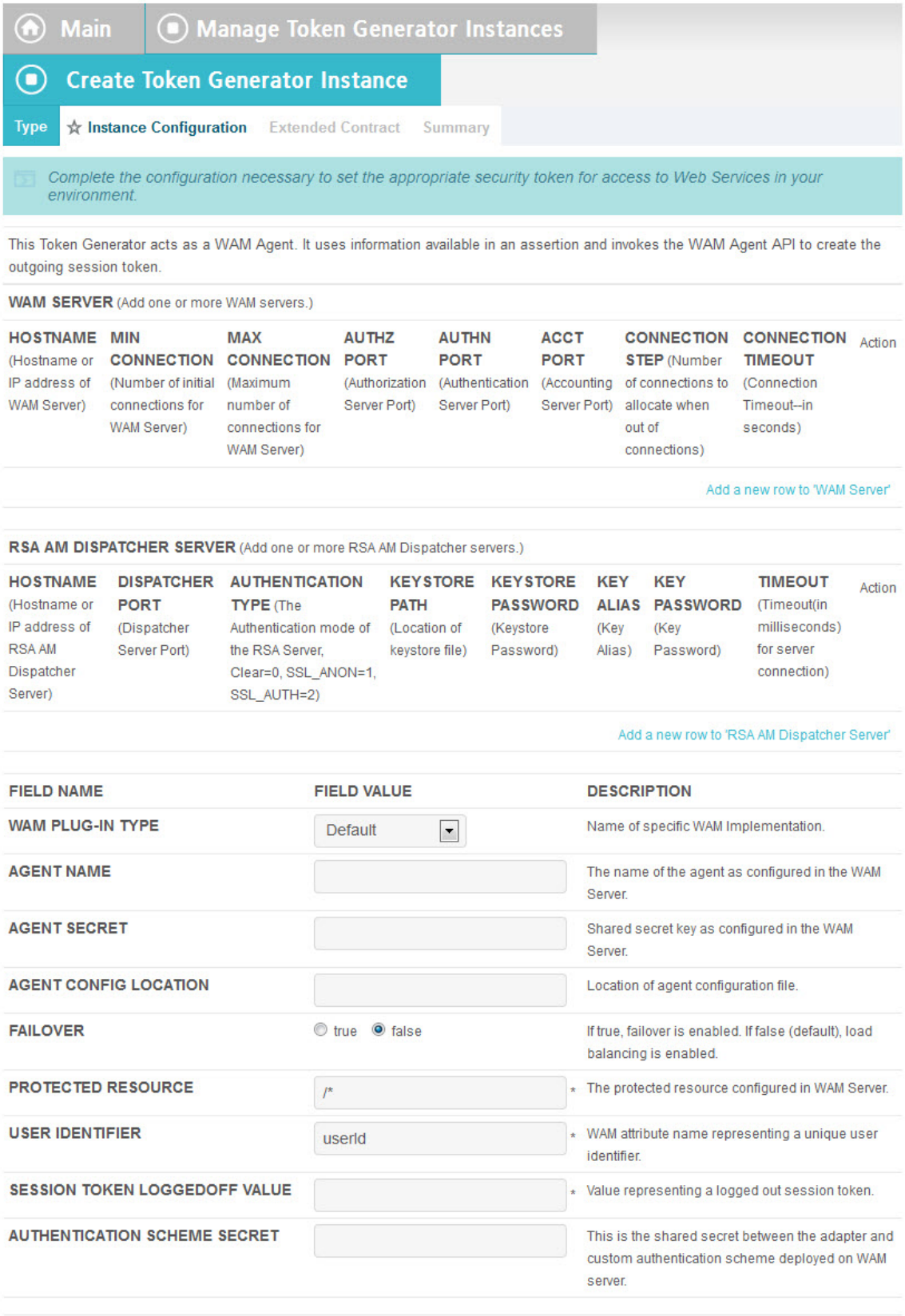
   Note: To enable token exchange, you may be prompted to provide SAML 1.x and SAML 2.0 federation identifiers for the STS on the Federation Info screen. Refer to the Federation Info screen's **Help** page for more information.

2. On the Manage Token Generator Instances screen, click **Create New Instance**.
3. On the Type screen, enter an Instance Name and Instance Id.

   The Name is any you choose for identifying this instance. The ID is used internally and may not contain spaces or non-alphanumeric characters.

4. Select WAM Token Generator 1.2 as the Type and click **Next**.

   Note: If you are configuring the adapter for a custom plug-in (not bundled with this kit), then continue to step 5. If you are configuring the RSA Dispatcher server, then continue with step 6. If you are configuring OAM, continue at step 7.


Main
Manage Token Generator Instances
Create Token Generator Instance
Type
Instance Configuration
Extended Contract
Summary
Complete the configuration necessary to set the appropriate security token for access to Web Services in your environment.
This Token Generator acts as a WAM Agent. It uses information available in an assertion and invokes the WAM Agent API to create the outgoing session token.
WAM SERVER (Add one or more WAM servers.)
HOSTNAME (Hostname or IP address of WAM Server)
MIN CONNECTION (Number of initial connections for WAM Server)
MAX CONNECTION (Maximum number of connections for WAM Server)
AUTHZ PORT (Authorization Server Port)
AUTHN PORT (Authentication Server Port)
ACCT PORT (Accounting Server Port)
CONNECTION STEP (Number of connections to allocate when out of connections)
CONNECTION TIMEOUT (Connection Timeout--in seconds)
Action
Add a new row to 'WAM Server'
RSA AM DISPATCHER SERVER (Add one or more RSA AM Dispatcher servers.)
HOSTNAME (Hostname or IP address of RSA AM Dispatcher Server)
DISPATCHER PORT (Dispatcher Server Port)
AUTHENTICATION TYPE (The Authentication mode of the RSA Server, Clear=0, SSL_ANON=1, SSL_AUTH=2)
KEYSTORE PATH (Location of keystore file)
KEYSTORE PASSWORD (Keystore Password)
KEY ALIAS (Key Alias)
KEY PASSWORD (Key Password)
TIMEOUT (Timeout(in milliseconds) for server connection)
Action
Add a new row to 'RSA AM Dispatcher Server'
FIELD NAME
FIELD VALUE
DESCRIPTION
WAM PLUG-IN TYPE
Default
Name of specific WAM Implementation.
AGENT NAME
The name of the agent as configured in the WAM Server.
AGENT SECRET
Shared secret key as configured in the WAM Server.
AGENT CONFIG LOCATION
Location of agent configuration file.
FAILOVER
true
false
If true, failover is enabled. If false (default), load balancing is enabled.
PROTECTED RESOURCE
/*
The protected resource configured in WAM Server.
USER IDENTIFIER
userId
WAM attribute name representing a unique user identifier.
SESSION TOKEN LOGGEDOFF VALUE
Value representing a logged out session token.
AUTHENTICATION SCHEME SECRET
This is the shared secret between the adapter and custom authentication scheme deployed on WAM server.

5. (Only for custom plug-ins for WAM servers other than OAM or RSA) On the Instance Configuration screen, click **Add a new row to 'WAM Server'** and provide the following information into the table:

   a. Enter the Hostname or the IP address where the WAM server is running.

   b. Specify the remaining WAM server values that are required for your configuration.

   c. Click **Update** in the Action column.

   d. Repeat this step as needed, for additional WAM plug-ins..

   Skip the next step.

6. (Only for the RSA bundled plug-in) On the Instance Configuration screen, click **Add a new row to 'RSA AM Dispatcher Server'** and provide the following information in the table:

   Note: You must specify at least one RSA AM Dispatcher Server

   e. Enter the Hostname or the IP address and the (optional) Dispatcher Port where the RSA AM Dispatcher server is running.

   Note: You must specify the authentication method that is used by the dispatcher server. If you have specified multiple dispatcher servers, each server can have individual authentication methods.

   f. Specify the Authentication Type used by the RSA Dispatcher Server.

      - **Clear** – clear text, no encryption
      - **Anon** – anonymous SSL, SSL encryption only
      - **Auth** – mutually authenticated SSL, SSL encryption with certificate-based encryption

   g. If the selected Authentication Type is **Auth**, you must specify the following RSA server values:

      - Keystore Path – String filename of the private Keystore file (PKCS12 only)
      - Keystore Password – password for the private Keystore
      - Key Alias – the alias to your private key in the Keystore
      - Key Password – private Key Password for Keystore

   h. (Optional) Specify the Timeout value required for your configuration.

   i. Click **Update** in the Action column.

   j. Repeat this step as needed for additional RSA Servers.

7. Provide entries on the Instance Configuration screen, as described on the screen and in the table below.

Note: The selected WAM Plug-in Type may override optional/required fields. For example, if the selected WAM Plug-n Type is `OAM`, the Agent Config Location becomes a required field. Leaving this field blank generates an error message.

Field	Description
WAM Plug-in Type	Class name for the specific WAM implementation. **Note:** WAM Plug Type determines optional/required fields.
Agent Name	This value must match the value used when the third-party WAM Web Agent was configured.
Agent Secret	This value must match the value used when the third-party WAM Web Agent was configured.
Agent Config Location	Required for OAM, this value must contain the full path to an XML network-configuration file generated by the access-management system.
Failover	The default is false, indicating load balancing is enabled and user-session states and configuration data are shared among multiple WAM servers. Select **true** to enable failover, indicating that when one server fails, the next server is used.
Protected Resource	(Required) All files in the root directory (/*) is the default. Specify a different path to the resources in the protected realm, if necessary.
User Identifier	(Required) Defines which attribute that is parsed from the WAM session token is the user identifier for use in the assertion.
Session Token LOGGEDOFF Value	(Required) Value representing a logged out session token.
Authentication Scheme Secret	(Required, except for RSA) The shared secret between the adapter and the custom authentication scheme deployed on the WAM server.

8. Click **Next**.
9. (Optional) On the Extended Contract screen, add attributes you expect to retrieve in addition to the SAML subject (user ID).

   (For more information on using the Extended Contract screen, see Extending an Adapter Contract in the PingFederate *Administrator's Manual.*)
10. Click **Next**.

11. On the Summary screen, verify that the information is correct and click **Done**.
12. On the Manage Token Generator Instances screen, click **Save**.

Creating a Custom Authentication Scheme for OAM

The Token Generator uses a custom authentication scheme when creating a WAM session and validates authentication requests coming from PingFederate. This section describes how to deploy the OAM-compatible Java-based PingFederate Custom Authentication Scheme.

1. From the `<token_translator_install_dir>/dist` directory, import the following file into the OAM:

   `PingCustomAuthPlugin.jar`

   The `PingCustomAuthPlugin.jar` file is a custom authentication scheme that supports OAM.

2. Configure your Access Server to use the custom authentication plug-in by creating or modifying a custom authentication scheme.

   Refer to Oracle Support documentation for additional information.

Note: The secret you specify when creating the custom authentication scheme must match the secret stored in the PingFederate Token Generator.

Using the STS Client SDK

Ping Identity provides a Java STS-Client SDK for enabling Web Service applications (Client or Provider) to interact with the PingFederate STS. (The SDK is available for download on the Ping Identity Downloads page (`www.pingidentity.com/products/downloads.cfm`).

The SDK provides functionality for sending a security token to the PingFederate STS for exchange with a returned SAML token, which can then be used to access Web Services across domains. The following code examples show how to send a token and request the exchange. Refer to the SDK documentation for modifications that apply to your site.

Java Sample Code

Token Processor

The code snippet below demonstrates using the PingFederate Java STS Client SDK to send a WAM session token to the PingFederate STS.

```
// Example method for obtaining the WAM Session token.
// You will need to implement this for your environment.
String wamSessionToken = getWAMSessionToken();

// Configure STS Client (IdP side / SP Connection)
```

```
STSClientConfiguration stsConfig = new STSClientConfiguration();
stsConfig.setAppliesTo("http://sp.domain.com");
stsConfig.setStsEndpoint("https://idp.domain.com:9031/idp/sts.wst");
stsConfig.setInTokenType(TokenType.BINARY);

// Instantiate the STSClient
STSClient stsClient = new STSClient(stsConfig);

// Send an RST Issue request to PingFederate STS
Element samlToken = stsClient.issueToken(wamSessionToken);
```

Token Generator

The code snippet below demonstrates using the PingFederate Java STS Client SDK to retrieve a WAM session token through the PingFederate STS.

```
// Configure STS Client (SP side / IdP Connection)
STSClientConfiguration stsConfig = new STSClientConfiguration();
stsConfig.setStsEndpoint("https://sp.domain.com:9031/sp/sts.wst");
stsConfig.setOutTokenType(TokenType.BINARY);

// Instantiate the STSClient
STSClient stsClient = new STSClient(stsConfig);

// Send an RST Issue request to PingFederate STS
Element wamsessionToken = stsClient.issueToken(samlToken);
```