PingFederate®

X.509 Token Translator

Version 1.0

User Guide

Part Number 3007-602
Version 1.0
April, 2009

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

Disclaimer

Contents

Introduction

The PingFederate X.509 Token Translator provides an Identity Provider (IdP) Token Processor for use with the PingFederate WS-Trust Security Token Service (STS). The Token Processor allows the STS to accept and validate an X.509 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).

Note: Ping Identity provides Java and .NET WSE3 STS-Client Software Development Kits (SDKs) for enabling Web Service applications (Client or Provider) to interact with the PingFederate STS. The SDKs are available for download at pingidentity.com/products/downloads.cfm.

The X.509 Token Processor uses the PingFederate security infrastructure for certificate validation and management. PingFederate validates the trust of all certificates. A certificate is trusted if the root certificate of the issuing Certificate Authority (CA) is imported into the PingFederate trusted certificate store or the CA is trusted by the Java Runtime Environment (JRE) in use.

System Requirements

PingFederate 6.0 or higher must be installed to implement the Token Processor.

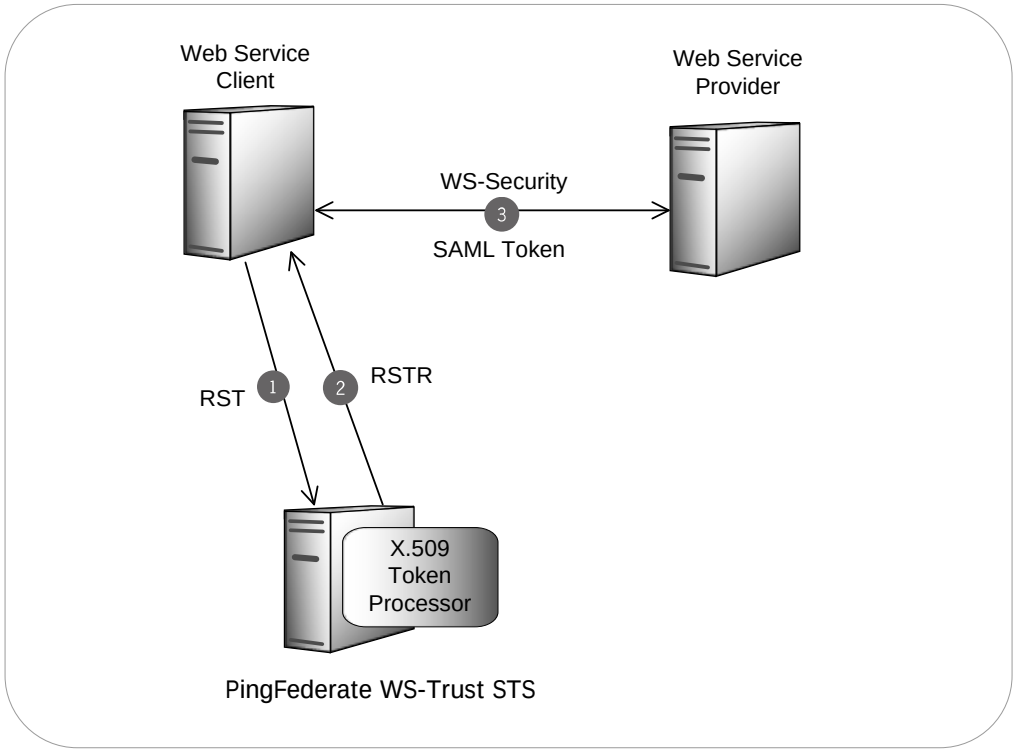
ZIP Manifest

The distribution ZIP file for the X.509 Token Translator contains the following:

- `/docs` – contains documentation:
  - `X509_STS_Token_Translator_Qualification_Statement.pdf` – testing and platform information
  - `X509_STS_Token_Translator_User_Guide.pdf` – this document
- `/dist` – contains libraries needed to run the token processor:
  - `pf-X509-token-translator-1.0.jar` – the X.509 Token Processor JAR file

WS-Trust STS Processing

The following figure shows a basic WS-Trust STS scenario in which PingFederate validates an X.509 token and issues a SAML token:



Processing Steps

1. A WSC sends a Request Security Token (RST) message containing an X.509 token to the PingFederate STS IdP endpoint.
2. The PingFederate X.509 Token Processor validates the X.509 token and, if valid, maps attributes from the X.509 token into a SAML token. PingFederate issues the SAML token based upon 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 (WSSE) header and sends this via a SOAP request to the Web Service Provider (WSP).

Installation and Setup

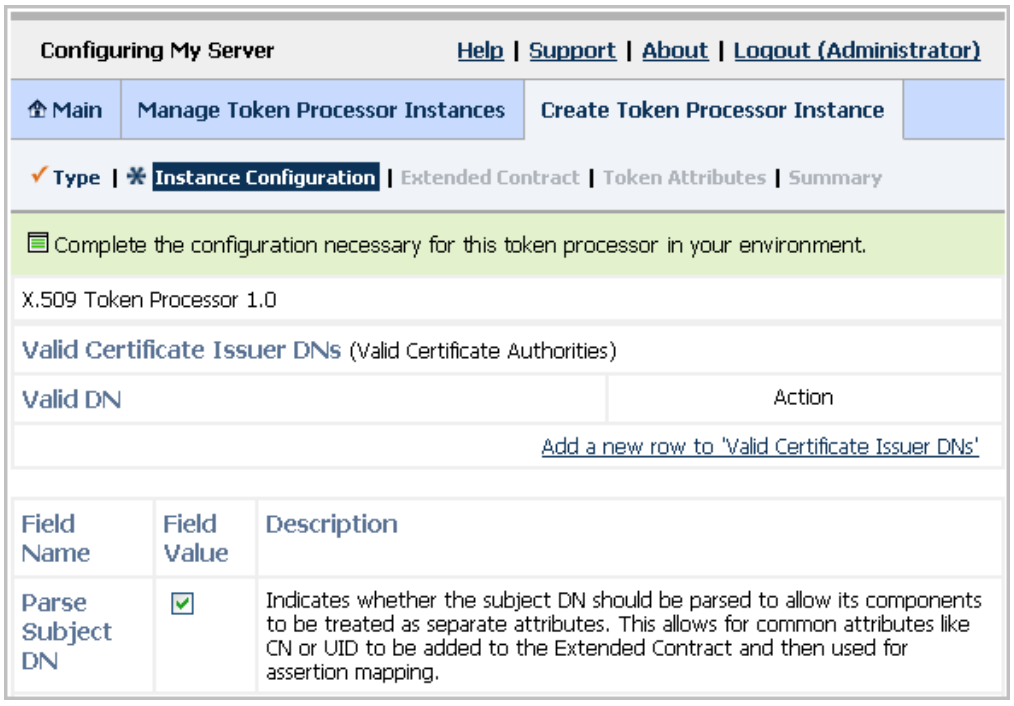
This section describes how to install and configure the X.509 STS Token Processor.

1. Copy the `pf-X509-token-translator-1.0.jar` file from the `dist` directory of this distribution to the `<pf-install>/pingfederate/server/default/deploy` directory of your PingFederate server installation.
2. In the `<pf-install>/pingfederate/bin` directory, open the file `run.properties` and change the value of `pf.secondary.https.port` from `-1` to a valid port number.
3. Log on to the PingFederate administrative console and click **Token Processors** under Application Integration Settings in the My IdP Configuration section of the Main Menu.

   If you do not see **Token Processors** on the Main Menu, enable WS-Trust by going to the Server Settings→Roles & Protocols screen and 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.

4. On the Manage Token Processor Instances screen, click **Create New Instance**.
5. On the Type screen, enter an Instance Name and Instance Id, and select X.509 Token Processor 1.0 as the Type.
6. Click **Next**.



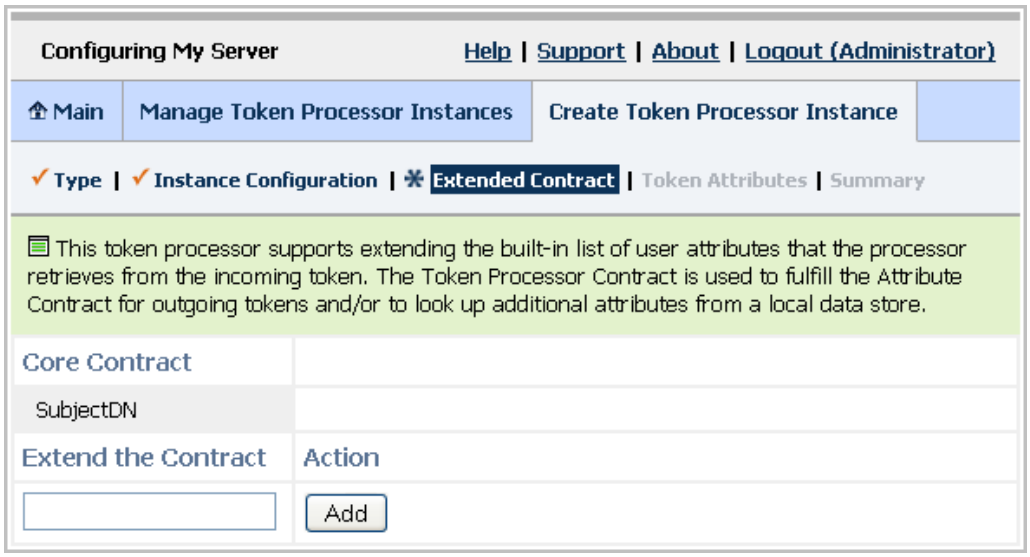
7. (Optional) Click **Add a new row to 'Valid Certificate Issuer DNs'** under Action, enter an applicable CA Issuer DN, and click **Update**.

   When this option is configured, only the CAs appearing under Valid DNs may be used to verify digital signatures on incoming tokens. Otherwise, all trusted CAs in the PingFederate or JRE trusted store may be used.

8. (Optional) If you *do not* want the adapter to parse the certificate Subject DN to make its elements available for the Extended Contract, clear the Parse Subject DN checkbox.

   Refer to the screen Description for more information.

9. Click **Next**.



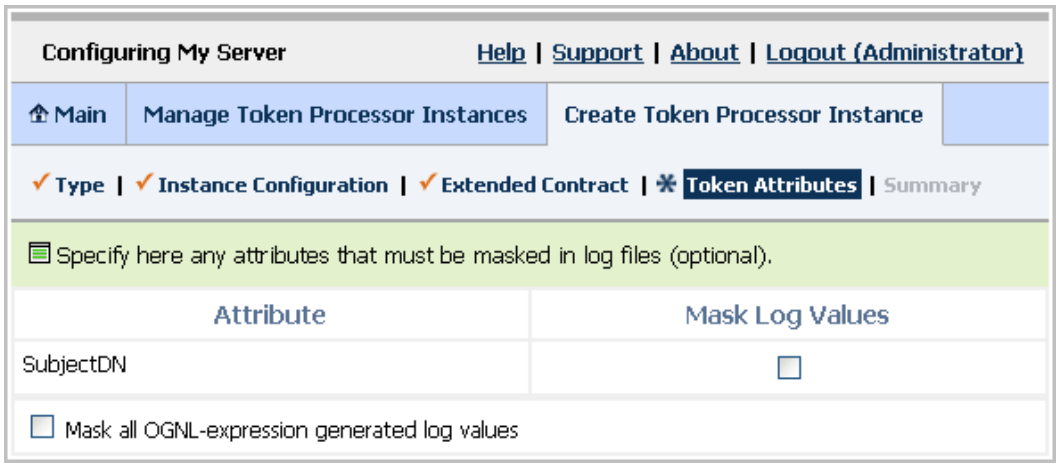
10. (Optional) On the Extended Contract screen, add any Subject DN components that you want to send individually in the SAML assertion, in addition to the full DN.

Note: The DN components are available separately only if the Parse Subject DN checkbox is selected (the default) on the Instance Configuration screen.

The attributes you enter must be in uppercase. Only attributes specified in RFC 2253 are allowed (CN, L, ST, O, OU, C, STREET, DC, and UID).

---

11. Click **Next**.



12. (Optional) On the Token Attributes screen, select any or all attributes whose values should be masked in PingFederate log files.

    Additionally, you may select **Mask all OGNL-expression generated log values**. (See the PingFederate *Administrator's Manual* for more information.)

13. Click **Next**.

14. On the Summary screen, verify that the information is correct and click **Done**.

15. On the Manage Token Processor Instances screen, click **Save**.

Using the STS Client SDKs

Ping Identity provides Java and .NET WSE3 STS-Client SDKs for enabling Web Service applications (Client or Provider) to interact with the PingFederate STS. (The SDKs are available for download at pingidentity.com/products/downloads.cfm.)

The SDKs provide 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

The code snippet below demonstrates using the PingFederate Java STS-Client SDK to send an X.509 token to the PingFederate STS:

```
// Example method for obtaining the X.509 token
// You will need to implement this for your environment
X500PrivateCredential credential = getCredentialFromKeystore();
```

```
// Configure STS Client (IdP side / SP Connection)
STSClientConfiguration stsConfiguration = new STSClientConfiguration();
stsConfiguration.setAppliesTo("http://sp.domain.com");
stsConfiguration.setStsEndpoint("https://idp.domain.com:9031/idp/sts.wst");
stsConfiguration.setInTokenType(STSClientConfiguration.TokenType.X509);

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

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

.NET Sample Code

The code snippet below demonstrates using the PingFederate .NET WSE3 STS-Client SDK to send an X.509 token to the PingFederate STS:

```
// Example method for obtaining the X.509 token
// You will need to implement this for your environment
X509Certificate2 certificate = new X509Certificate2(
("C:\\Temp\\cert.p12","test");

// Configure STS Client (IDP-side, SP connection)
STSClientConfiguration stsConfig = new STSClientConfiguration();
stsConfig.appliesTo = "http://sp.domain.com";
stsConfig.stsEndpoint = new Uri("https://idp.domain.com:9031/idp/sts.wst");

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

// Send RST Issue request to PingFederate STS
SecurityToken samlToken = stsClient.IssueToken(certificate);
```