PingFederate®

Apache Integration Kit

Version 3.1

For Linux

User Guide

Ping Identity

PingFederate Apache Integration Kit *User Guide*
Version 3.1
February, 2016

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: **February 29, 2016**

Contents

Introduction

The PingFederate Apache Integration Kit for Linux adds a Service Provider (SP) Web-server integration option to PingFederate by providing an Agent for the Web server. The Apache Agent works in conjunction with the PingFederate OpenToken Adapter to allow an SP enterprise to accept SAML assertions and provide single sign-on (SSO) to Apache HTTP Server Web applications. The assertions may be sent from an Identity Provider (IdP) using the SAML protocol (version 2.0 or 1.x) or the WS-Federation passive-requestor protocol (see Supported Standards in *Getting Started*).

Intended Audience

This document is intended for system administrators with experience in the configuration and maintenance of Linux/Unix and Apache HTTP servers. Knowledge of networking and user-management configuration is assumed. Please consult the documentation provided with your server tools if you encounter any difficulties in areas not directly associated with PingFederate or the Apache Agent.

ZIP Manifest

The distribution ZIP file for the Apache Integration Kit for RHEL contains the following:

- `ReadMeFirst.pdf` – Contains links to this online documentation.
- `/dist` – contains libraries needed for the Adapter and Apache Agent:
  - `opentoken-adapter-2.5.1.jar` – OpenToken Adapter JAR file
  - `opentoken-agent-2.5.1.jar` – OpenToken Agent JAR file
  - `/apache-agent` – the PingFederate Apache Agent
    - `/config`
      - `mod_pf.conf` – Apache Agent module configuration file
      - `start_page_template.html` – Apache Agent start-page template
      - `error_page_template.html` – Apache Agent error-page template
    - `/lib/Apache_2.2`
      - `/RHEL5_64` – for RHEL 5.x 64-bit
        - `libopentoken.so` – OpenToken library
        - `mod_pf.so` – Apache Agent module
      - `/RHEL6_64` – for RHEL 6.x 64-bit
        - `libopentoken.so` – OpenToken library
        - `mod_pf.so` – Apache Agent module
      - `/Ubuntu12_64` – for Ubuntu 12.04 64-bit
        - `libopentoken.so` – OpenToken library

        - `mod_pf.so` – Apache Agent module

– `/lib/Apache_2.4`

- `/RHEL6_64` – for RHEL 6.x 64-bit
    - `libopentoken.so` – OpenToken library
    - `mod_pf.so` – Apache Agent module
- `/RHEL7_64` – for RHEL 7.x 64-bit
    - `libopentoken.so` – OpenToken library
    - `mod_pf.so` – Apache Agent module
- `/Ubuntu12_64` – for Ubuntu 12.04 64-bit
    - `libopentoken.so` – OpenToken library
    - `mod_pf.so - Apache Agent module`
- `/Ubuntu14_64` – for Ubuntu 14.04 64-bit
    - `libopentoken.so` – OpenToken library
    - `mod_pf.so` – Apache Agent module

System Requirements

This PingFederate Apache Agent is designed and supported for Apache HTTP Server 2.2 and 2.4 (http://httpd.apache.org/) on RHEL 5, RHEL 6, RHEL 7 (2.4 only), Ubuntu 12.04 and Ubuntu 14.04 (2.4 only). The Agent supports 64-bit RHEL, Ubuntu operating systems, and both pre-fork and worker multi-processing modules.

The following additional prerequisites must be satisfied in order to implement the Apache Agent:

- PingFederate 6.x (or higher)

  Note: If not already done, ensure that all PingFederate deployments are updated with version 2.5.1 (or higher) of the OpenToken Adapter, supplied as part of this integration-kit distribution.

- OpenSSL 1.0.1g (or higher)

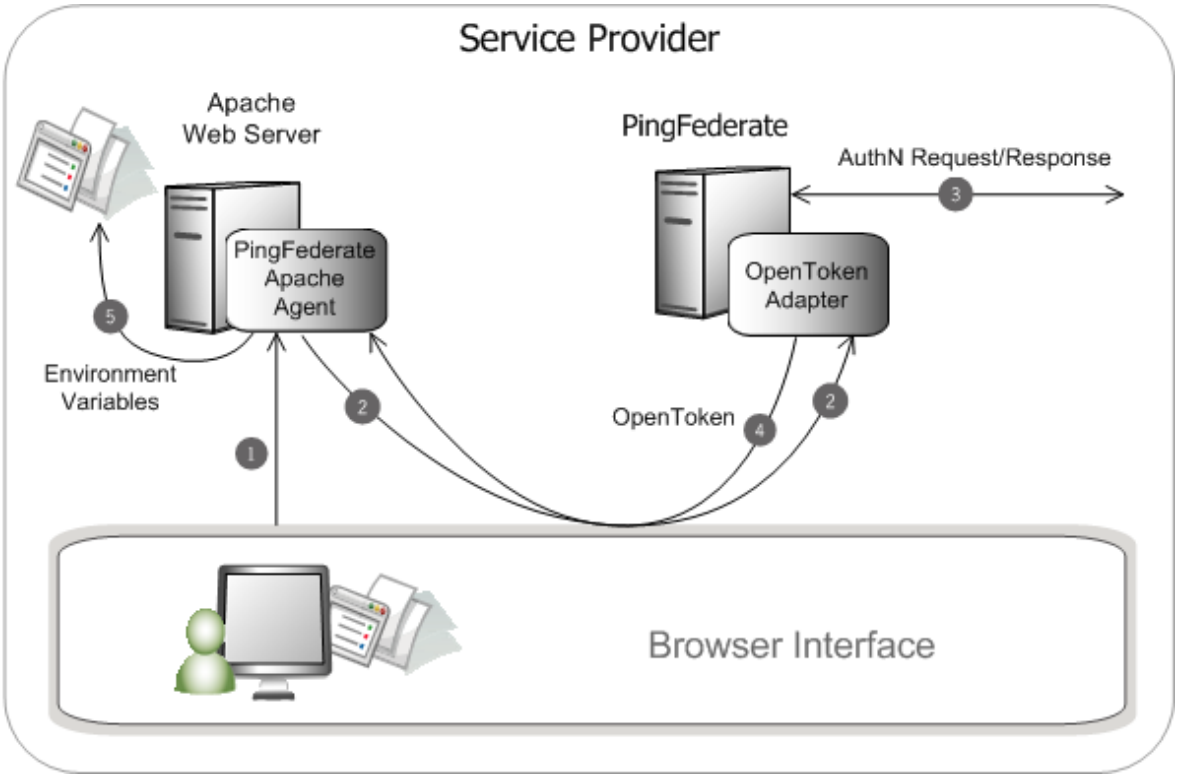
Processing Overview

The Apache Agent acts as a filter in front of an application (or any external protected resource). The basic responsibilities of the Agent are to filter requests to determine if a request is for a protected resource:

- If the request is for an unprotected resource, the Agent passes the request to the application.
- If the request is for a protected resource, the Agent checks to see if there is a PingFederate session available and if the session parameters meet session policy for that session.
- If a session exists and the session meets session policy for that request, the Agent passes the request through to the application.

- If a session does not exist, or if the existing session does not meet the session policy for that request, the Agent redirects the user's browser through the PingFederate server to an IdP for authentication. After authentication, PingFederate redirects the user back to the protected resource with a valid session.

The following figure illustrates an SP-initiated SSO scenario, showing the request flow and how the PingFederate OpenToken Adapter wraps attributes from an assertion into a secure token (`OpenToken`) and passes the token to Apache.



Processing Steps

1. A user attempts to access a resource on the Apache server protected by the PingFederate Apache Agent.

   The user is redirected to the PingFederate server for authentication.

   (If an `OpenToken` session already exists, the user is granted immediate access.)

2. The PingFederate server redirects the user's browser to an IdP for authentication using either the SAML or WS-Federation protocols. The IdP partner authenticates the user and returns a SAML assertion.

3. PingFederate validates the assertion and creates an `OpenToken` for the user including any configured attributes. PingFederate then redirects the browser, including the `OpenToken`, back to the Apache Agent.

4. The Agent verifies the `OpenToken` and grants access to the protected resource. The User ID and any attributes from the `OpenToken` are exposed to the resource as HTTP Request Headers or Apache Environment Variables.

Installation and Setup

Setting up the PingFederate Apache Integration Kit involves:

- Installing and configuring the OpenToken Adapter in PingFederate
- Installing and configuring the Apache Agent

Installing the OpenToken Adapter and Configuring PingFederate

Note: If you have already deployed version 2.5.1 (or higher) of the OpenToken Adapter, skip steps 1 through 4 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`

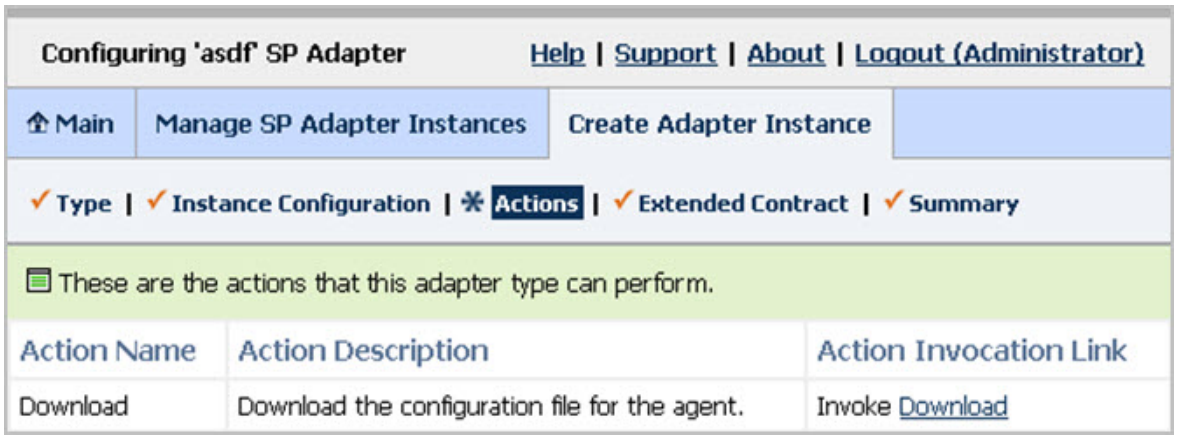
   The adapter JAR file is `open-token-adapter-<version>.jar`.
3. Unzip the integration-kit distribution file and copy `opentoken-adapter-2.5.1.jar` from the `/dist` directory to the PingFederate directory:

   `<PF_install>/pingfederate/server/default/deploy`
4. Start or restart PingFederate.
5. Create an instance of the OpenToken Adapter for your SP configuration using settings on the Instance Configuration screen as indicated in the table below. (Fields not specified are optional. For more information, see OpenToken Adapter Configuration in the PingFederate *Administrator's Manual*.)

Option	Description
Password	Enter any password you choose.
Confirm Password	Password confirmation

6. On the Actions screen, click the **Download** link and then click **Export** to save the properties file to your file system.

Note: If you have access to the Apache server, you can download the file directly to the server's `conf` directory, where the PingFederate Apache Agent looks for the properties by default. The path and filename are configurable (see Configuring the Apache Agent).



7. On the Extended Contract screen, enter any attributes you want to pass to the application as HTTP request headers or environment variables.

   For more information, see the section About Session Information.

8. Configure or modify the connection(s) to your IdP partner(s) using the OpenToken Adapter you configured in the last steps.

   For more information, see Identity Provider SSO Configuration in the PingFederate *Administrator's Manual*.

Installing and Configuring the Apache Agent

To set up the Apache Agent, you must copy a configuration file and a library from the extracted integration-kit distribution to your Apache installation and then modify the configuration file.

> **Note:** The Agent setup is required for initial installations and for upgrades. We strongly recommend reinstalling the PingFederate Apache Agent in your Apache installation.

Deploying the Agent

The PingFederate Apache Agent is represented by the `mod_pf.so` Apache module (dynamic library) and an auxiliary OpenToken library. The behavior of the Agent is controlled by properties contained in the `mod_pf.conf` file.

To deploy the Agent:

1. Unzip the integration-kit distribution file and copy the contents of the `/dist/apache-agent/lib` directory that corresponds to your version of Linux into your Apache `/modules` directory.

   Be sure to copy both shared-object files for the indicated platform.

2. For new installations, from the integration-kit `/dist/apache-agent/config` directory, copy the `mod_pf.conf`, `start_page_template.html` and the `error_page_template.html` into the `/conf` directory of your Apache installation.

   The `mod_pf.conf` file must be configured: see Configuring the Apache Agent.

3. Ensure that the properties file downloaded during the PingFederate adapter setup (default name: `agent-config.txt`) is available to the Agent by placing it into Apache's `/conf` folder (see step 6).

4. If you are using Security Enhanced Linux, run the following commands, as `root`, to allow the agent to run in the `httpd` context:

```
chcon --reference /usr/sbin/httpd /etc/httpd/modules/mod_pf.so
chcon --reference /usr/sbin/httpd /etc/httpd/modules/libopentoken.so
```

Note: The paths above assume the default Linux installation.

5. Ensure the following module is enabled in `httpd.conf`:

   Apache 2.2:

```
LoadModule authz_host_module modules/mod_authz_host.so
```

   Apache 2.4:

```
LoadModule access_compat_module modules/mod_access_compat.so
```

6. Add the following directives to `httpd.conf`. These directives must appear after the `LoadModule` statement listed in the previous step:

```
LoadFile        modules/libopentoken.so
LoadModule      pf_module modules/mod_pf.so
PingFederateConfigurationFile    conf/mod_pf.conf
```

Caution: Ensure these directives are in the order illustrated above.

7. Add the following directive within all `Directory` contexts that should be handled by the Agent:

```
AuthType PFApacheAgent
```

   We highly recommend using a deny by default configuration for all directories which are to be protected by the Agent. So in addition to the above AuthType we recommend these settings:

```
Order Deny,Allow
Deny from all
```

8. Restart Apache.

Configuring the Apache Agent

You must modify the file `mod_pf.conf` for your environment. Refer to comments in the file (located in the `conf` directory of your Apache installation) and configure the required properties—modify "example" placeholders, at minimum, as well as required and optional defaults as needed.

Note: Changes to `mod_pf.conf` do not take effect until the server is restarted.

Configuring Virtual Hosts

Each virtual host can optionally have it's own respective Agent configuration. To use this feature add the following parameter within the virtual host context:

```
PingFederateConfigurationFile conf/mod_pf_vhost.conf
```

In this example `mod_pf_vhost.conf` is an agent configuration file that contains settings unique to the virtual host. PingFedederate attributes from the base server configuration are not merged with the virtual host when the PingFederateConfigurationFile attribute is specified for the virtual host.

Note: If no custom configuration is provided a virtual host will use the agent configuration from the base server.

Testing the Apache Agent

The Apache Agent includes a protected start page for testing to verify the installation and configuration. The start page initiates an SSO transaction with the IdP partner, and if successful, displays the HTTP headers that the Apache Agent exposes to an underlying application. These headers correspond to attributes from the SAML assertion.

Access the Apache Agent protected start page at:

```
http://<apache-server>/<protected-path>/?cmd=PingStartPage
```

Caution: This feature is for testing and demonstration only. To enable this feature, comment out the `PingFederateStartPageURL` property in the file: `<apache_home>/conf/mod_pf.conf`.

For security reasons, this feature is disabled by default and should remain disabled in a production environment.

Agent Runtime Information

The following sections provide additional information about the behavior and functionality of the PingFederate Apache Agent.

Session Validation

PingFederate validates both an *inactivity timeout* and an *overall session timeout*:

- The inactivity timeout is the amount of time that a session can be inactive (i.e., during which no new browser requests are received) before a user is required to re-authenticate.
- The overall session timeout is the total amount of time that a session can be active, regardless of activity, before the user is required to re-authenticate.

If either of the session limits has expired, the Apache Agent cancels the existing session and redirects the user to the configured `PingFederateLoginPageUrl` to start an SP-initiated SSO for authentication at an IdP.

Note: Session cancellation enforces session cleanup in the PingFederate server and obsolescence of session cookies.

Memory Usage

The Apache Agent uses the Apache Portable Runtime (APR) pools for allocation of most data: either the configuration-time pool for storing configuration variables or request-time pools for processing sessions. Heap is used only for temporary data with a short usage time and sensitive size—for instance, for dynamic reallocations on compressed-token decompression, parsing session information, or operations with the `OpenToken` library.

About Session Information

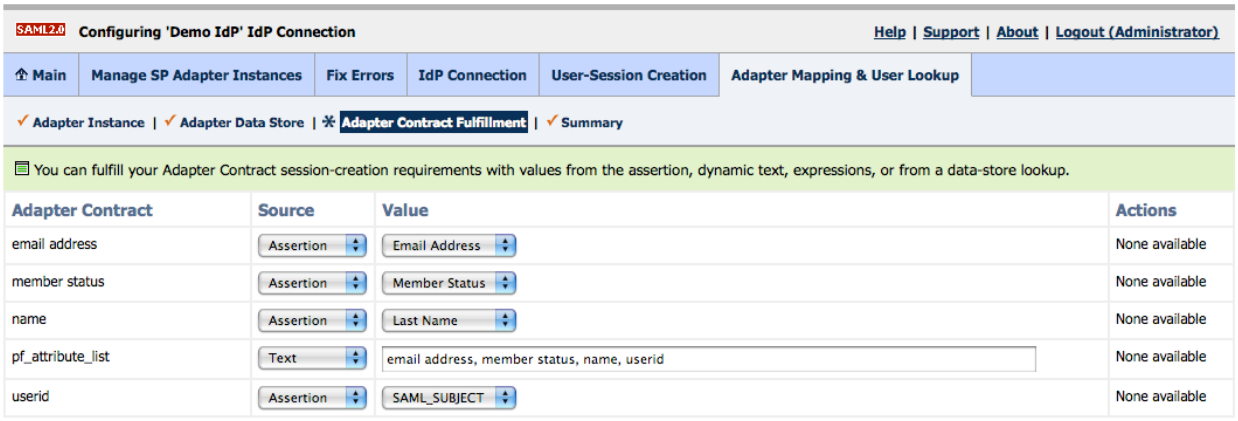
The PingFederate Apache Agent exposes session information and user attributes from the adapter to the protected application via HTTP request headers or Apache environment variables. This information can then be used by the application for authorization decisions, for example, or for generation of content specific to the user making the request.

The session and attribute information exposed to the application includes the following:

- Attributes from the OpenToken Adapter contract – These include, by default, the subject (`SUBJECT`) and attributes specified on the Extended Contract screen of the adapter setup. Only the attributes fulfilled at runtime are exposed to the application; attributes with a `NULL` value are not included in the OpenToken.
- `NOT-ON-OR-AFTER` – The time until inactivity timeout is reached.
- `RENEW-UNTIL` – The time until overall session timeout is reached.
- `AUTH_NOT-BEFORE` – The time when the session was created.
- `AUTHNCONTEXT` – Information from the SAML assertion that describes how the user was authenticated at the IdP.

For security reasons, each HTTP request header or Apache environment variable is first pre-pended with a specific (configurable) prefix (see Configuring the Apache Agent). The Apache Agent always removes and rewrites these prefixed request headers and/or environment variables for each request.

If applications protected by the Apache Agent cannot be modified to accept headers with this prefix, the Apache Agent can be configured not to add a prefix to the HTTP headers and/or environment variables. In this case, on the Extended Contract screen in the OpenToken Adapter configuration, include an attribute named `pf_attribute_list`. Then map that attribute in your IdP Connection as a Text field containing a comma-separated list of all the attributes in the adapter contract (see figure below). This attribute list is sent in the OpenToken and used by the Apache Agent to overwrite headers in the request.



For more information, see Configuring Adapter Contract Fulfillment in the PingFederate *Administrator's Manual*.

Logging

The PingFederate Apache Agent uses a standard Apache API logging scheme that writes into the standard `logs/error_log` file. This file is created automatically at startup (if it is absent) with the verbosity level controlled by a standard option `LogLevel` in `httpd.conf`. Additionally, the PingFederate Apache Agent has six internally distinguished verbosity levels, ranging from 0 to 5. The first four correspond to Apache definitions in `error/warn/notice/info`. The last two levels are for logging HTTP requests/responses and cURL-library debug output, if necessary. The default level is 0, which logs only errors.

> **Note**: The Apache Agent logs all of its output at the `info` level. To have access to this output, set the Apache `LogLevel` to `info` in `httpd.conf`. You must re-start the Apache server after changing the configuration file.

SSL Support

The Apache Agent supports TLS/SSL, using standard Apache SSL support for connections to the server from browsers.

Error Handling

In case of errors, the Apache Agent redirects the inbound request to a configured error-page URL (see Configuring the Apache Agent).

Session Logout

The PingFederate Apache Agent provides a configurable URL (`PingFederateCancelURL`) for user-session logout (see Configuring the Apache Agent). This URL specifies a resource that directs the Apache Agent to initiate an SLO first, if configured to do so, and then expire the PingFederate session, clean up any resources associated with the session, and pass control back to the application so that it can clean up its own session.