



Agentless Integration Kit

Version 1.0

User Guide

PingIdentity®

© 2010 Ping Identity® Corporation. All rights reserved.

PingFederate Agentless Integration Kit *User Guide*
Version 1.0
October, 2010

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: www.pingidentity.com

Trademarks

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

All other trademarks or registered trademarks are the properties 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

Introduction.....	4
Intended Audience	4
Additional Resources	4
ZIP Manifest	5
System Requirements.....	5
Installation.....	5
IdP Overview and Configuration.....	5
IdP Process Overview.....	5
Configuring the ReferenceID IdP Adapter	7
SP Overview and Configuration.....	10
SP Process Overview	10
Configuring the SP Adapter	11
Using Mutual SSL/TLS Authentication	14
Application Integration	15
Authenticating to PingFederate.....	15
Application Endpoints.....	15
Using HTTPS	15
Attribute Formatting.....	15
Sample Java Code.....	16

Introduction

The PingFederate Agentless Integration Kit includes the ReferenceID Adapter, which allows developers to integrate applications with a PingFederate server acting as either an Identity Provider (IdP) or a Service Provider (SP). The ReferenceID Adapter allows an IdP server to receive user attributes from an IdP application. On the SP side, the adapter allows an SP application to receive user attributes from the SP server.

The ReferenceID Adapter does not require the application to include *agent* PingFederate software libraries to interact with the Adapter. Instead, user attributes are passed via direct HTTP calls between the application and PingFederate.

For IdP integration, after user authentication, the application makes a direct HTTP call to PingFederate with user attributes, which PingFederate temporarily stores, sending a reference to them in the HTTP response. The IdP application redirects the browser to PingFederate, including the reference.

For SP integration, PingFederate parses the SAML assertion and temporarily stores the user attributes, generating a reference to them and sending the reference in a redirect to the SP application. The application makes a direct HTTP call back to PingFederate with the reference, and PingFederate returns the attributes in the HTTP response.

Intended Audience

This document is intended for system administrators with some knowledge of PingFederate, and for Web-application developers with a working knowledge of Internet user authentication and HTTP transport methodology.

Additional Resources

Administrators may want to review the PingFederate *Administrator's Manual*—specifically the information on adapters and integration kits.

Tip: If you encounter any difficulties with configuration or deployment, please try searching the Ping Identity [Customer Portal](http://www.pingidentity.com/support-and-downloads/portal.cfm) (www.pingidentity.com/support-and-downloads/portal.cfm) under **Answers**.

ZIP Manifest

The distribution ZIP file for the Integration Kit contains:

- `/dist` – contains the Java libraries:
 - `pf-referenceid-adapter-1.0.jar` – ReferenceID Adapter JAR file
 - `json_simple-1.1.jar` – JavaScript Object Notation (JSON) JAR file, used for attribute formatting (see “[Attribute Formatting](#)” on page 15).
- `/docs` – contains this documentation:
 - `Agentless_Integration_Kit_Qualification_Statement.pdf` – testing and platform information
 - `Agentless_Integration_Kit_User_Guide.pdf` – this document

System Requirements

The ReferenceID Adapter requires installation of PingFederate 5 or higher.

Installation

To install the ReferenceID Adapter:

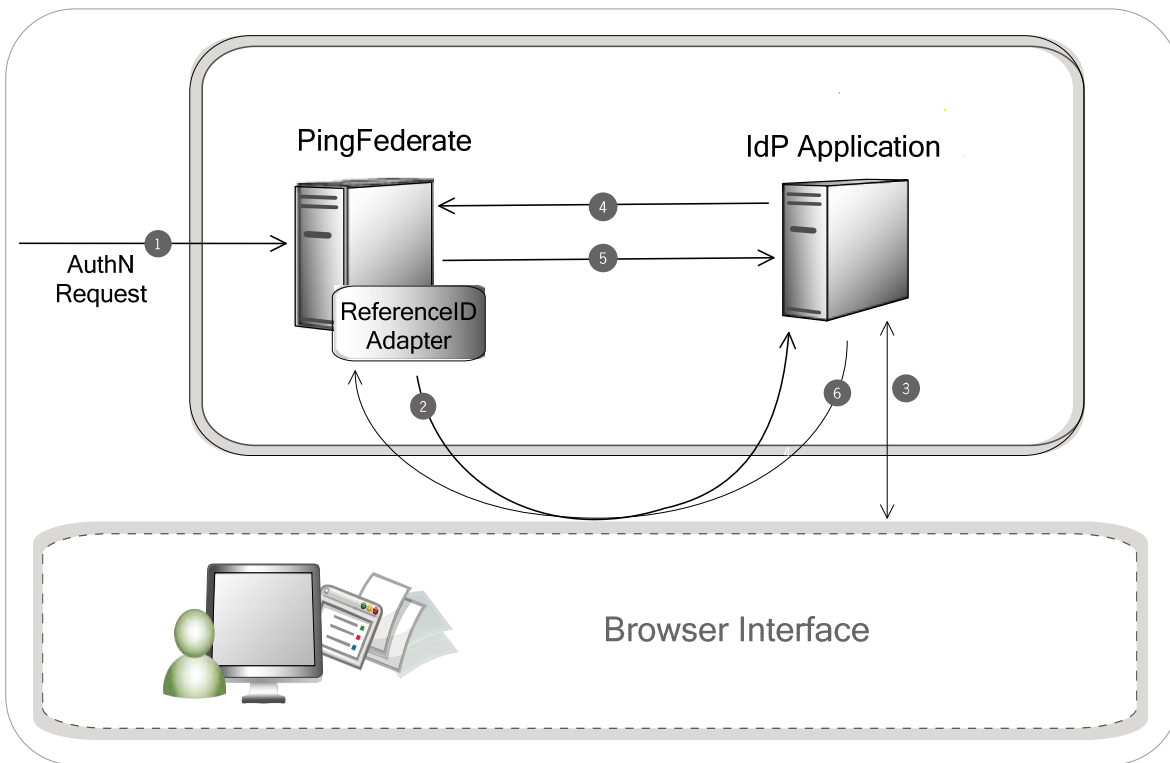
1. From the integration-kit `dist` directory, copy *both* the `pf-referenceid-adapter-1.0.jar` and the `json_simple-1.1.jar` into:
`<PF-install>/server/default/deploy`
2. Start or restart PingFederate.

IdP Overview and Configuration

This section provides an overview of SSO processing using the IdP ReferenceID Adapter as well as instructions for setting up the Adapter in PingFederate.

IdP Process Overview

The following figure displays an example SSO (SP-initiated) process flow between PingFederate and the IdP application using the ReferenceID Adapter:



Processing Steps

1. PingFederate receives an authentication request for a SAML assertion.
2. The PingFederate server redirects the browser to the IdP application for authentication, including the resume path as a query parameter.

The server needs the resume path back from the application to continue SSO processing after the user authenticates.

3. The IdP application authenticates the user.
4. The IdP application uses a direct HTTP call to send the user attributes (as JSON-encoded objects) to PingFederate. For example:
5. PingFederate stores the attributes and returns a reference in the HTTP response to the IdP application.
6. The IdP application redirects the browser to the PingFederate resume path (received in Step 2) with the reference in the query string. For example:

```
https://pingfederate.example.com:9031/ext/ref/dropoff
```

```
https://pingfederate.example.com:9031/[resume-path]?REF=EFG123
```

7. (Not shown) PingFederate creates a SAML assertion using the attributes associated with the ReferenceID and sends the assertion to the Service Provider.

Configuring the ReferenceID IdP Adapter

To configure the IdP Adapter:

1. Log on to the PingFederate administrative console.
2. Click **Adapters** under My IdP Configuration on the Main Menu.
(For more information about IdP adapters, see the PingFederate *Administrator's Manual*.)
3. Click **Create New Instance** on the Manage IdP Adapter Instances screen.

Note: References to screens in these steps conform to the appearance of the PingFederate 6.x administrative console. However, the configuration is the same for previous versions; only the screen names have changed.

4. On the Type screen, enter an Instance Name and Instance Id.

The Name is any you choose for identifying this Adapter Instance elsewhere in the administrative console. The ID is used by PingFederate internally and may not contain spaces or non-alphanumeric characters. Both must be unique among other configured adapter instances.

5. Select **ReferenceID Adapter 1.0** from the Type list and click **Next**.

Configuring IdP Adapter

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

Main | Manage IdP Adapter Instances | Create Adapter Instance

✓ Type | * IdP Adapter | Actions | Extended 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.

The ReferenceID Adapter allows user attributes to be passed in and out of the PingFederate server via direct HTTP(S) calls. Attributes are retrieved via a ReferenceID.

Field Name	Field Value	Description
Authentication Endpoint	<input type="text" value="https://"/> *	Application endpoint URL where the end user is redirected for authentication.
User Name	<input type="text"/> *	ID the application uses to authenticate to the PingFederate server.
Pass Phrase	<input type="text"/> *	Pass phrase the application uses to authenticate to the PingFederate server.
Allowed Subject DN	<input type="text"/>	Subject DN from the client certificate. If entered, PingFederate restricts client-certificate authentication (when enabled) by matching against this DN.
Allowed Issuer DN	<input type="text"/>	Issuer DN from the client certificate. If entered, PingFederate restricts client-certificate authentication (when enabled) by matching against this DN.
Logout Service Endpoint	<input type="text"/>	Application endpoint URL used for single logout.

Show Advanced Fields

6. Provide entries on the IdP Adapter screen, as described on the screen and in the table below.

Field Name	Description
Authentication Endpoint	Enter the application URL to which PingFederate redirects the end user for authentication.
User Name	Enter an ID for the application to use for authentication.
Pass Phrase	Use the next screen to display the clear-text value of the pass phrase you enter here, for copying to the application.
Allowed Subject DN	(Optional) Client-certificate authentication may be used in addition to Basic authentication (User Name and Pass Phrase). To enable client-certificate-authentication, specify the Subject DN of the client certificate. Both Subject DN and Issuer DN must be specified to enable client-certificate authentication. Note: For information about configuring PingFederate to use this form of authentication in certain cases, see “Using Mutual SSL/TLS Authentication” on page 14.
Allowed Issuer DN	(Optional) To enable client-certificate authentication, specify the Issuer DN of the incoming client certificate. Both Subject DN and Issuer DN must be specified to enable client certificate authentication. (See the Note above.)
Logout Service Endpoint	(Optional) Enter the IdP-application URL where the user can initiate SAML single logout (SLO). SLO allows a user to log out of both the IdP and the SP sites with one action (for more information, see “Supported Standards” in the PingFederate manual <i>Getting Started</i>). For more information, see Logout Mode in the table for Advanced Fields under the next step.

7. (Optional) Click **Show Advanced Fields** to view additional configuration settings.

You can change default values or settings, depending on your network configuration and other requirements at your site:

Field Name	Description
Reference Duration	PingFederate caches the reference and attributes for this minimum period of time. This field is provided for administrators to make adjustments, as needed, to address network latency issues.
Reference Length	Increasing the length of the reference makes it more difficult to replicate when security is a concern.
Require SSL/TLS	(Optional) We recommend using the secure transport protocol unless a secure, dedicated network segment exists between the application server and PingFederate.

Field Name	Description
Outgoing Attribute Format	As an option, you can change the format in which PingFederate encodes attribute values on the HTTP response to the application (see “ Attribute Formatting ” on page 15).
Incoming Attribute Format	As an option, you can change the format in which the application encodes attribute values on the HTTP request to PingFederate (see “ Attribute Formatting ” on page 15).
Logout Mode	Use these options to define how to handle application logout. Front Channel (the default) redirects the browser to the application endpoint, including the reference as a query parameter. When resolved this reference gives all of the user attributes as well as the resume path for the application to use in a logout response. Back Channel sends a direct HTTP request from the server to the application. The variable <code>\${attribute-name}</code> may be used for any attribute to build a dynamic URL.
Skip Host Name Validation	(Optional) Select the check box to skip host name validation, for example, when testing or when the host name validation cannot be performed.

8. Click **Next**.

9. (Optional) On the Actions screen, click **Show Pass Phrase**.

Use this option to copy and paste the pass phrase into the application to facilitate HTTP Basic authentication between the application and the PingFederate server.

10. Click **Next**.

11. (Optional) On the Extended Contract screen, add attributes you expect to retrieve in addition to the SAML subject (user ID).

(For more information about this screen, see the *PingFederate Administrator's Manual* or click **Help**.)

12. On the Adapter Attributes screen, select the **Pseudonym** check box to send a pseudonym to the SP that uniquely identifies a user for account linking.

(For more information about this screen, see the *PingFederate Administrator's Manual* or click **Help**.)

(Optional) Select the **Mask Log Values** check box for each attribute you want to mask in the log file.

Note: If OGNL expressions might be used to map derived values into outgoing assertions and you want those values masked, select the associated check box below the Attribute list. (For more information, see “Using Attribute Mapping Expressions” in the *PingFederate Administrator's Manual*.)

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

14. On the Manage IdP Adapter Instances screen, click **Save** to complete the Adapter configuration.

15. Configure or modify the connection(s) to your SP partner(s) using the ReferenceID Adapter Instance.

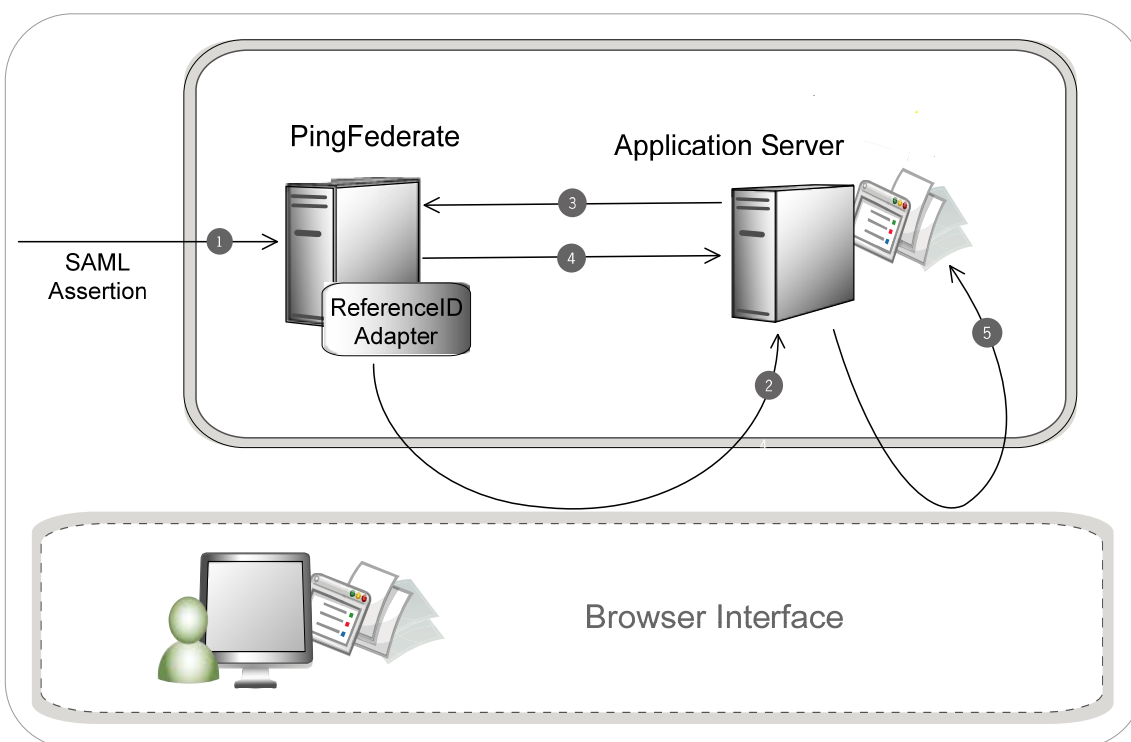
(For more information, see the “Identity Provider SSO Configuration” chapter in the PingFederate *Administrator’s Manual*.)

SP Overview and Configuration

This section provides an overview of SSO processing using the SP ReferenceID Adapter as well as instructions for setting up the Adapter in PingFederate.

SP Process Overview

The following figure displays a typical SSO process flow between PingFederate and the SP application using the ReferenceID Adapter:



Processing Steps

1. PingFederate receives a SAML assertion from an IdP partner. The assertion is validated and parsed into the user attributes, which are temporarily maintained within PingFederate.
2. The PingFederate server redirects the user to the target application with a reference to the user attributes. The reference is included in the URL query string. For example:
`https://target.example.com?REF=ABC123`

3. The target application makes an authenticated direct HTTP(S) call to PingFederate to retrieve the user attributes. For example:
`https://pingfederate.example.com:9031/ext/ref/pickup?REF=ABC123`
4. PingFederate looks up the attributes (in the above example, referenced by ABC123) and provides them to the target application in the HTTP response.
5. The target application uses the attributes to create a user session, enabling access to the target resource.

Configuring the SP Adapter

To configure the SP Adapter:

1. Click **Adapters** under My SP Configuration on the Main Menu.

(For more information about SP adapters, see the PingFederate *Administrator's Manual*.)

2. Click **Create New Instance** on the Manage SP Adapter Instances screen.
3. On the Type screen, enter an Instance Name and Instance Id.

The Name is any you choose for identifying this Adapter Instance elsewhere in the administrative console. The ID is used by PingFederate internally and may not contain spaces or non-alphanumeric characters. Both must be unique among other configured adapter instances.

4. Select **ReferenceID Adapter 1.0** from the Type list and click **Next**.

Note: References to screens in these steps conform to the appearance of the PingFederate 6.x administrative console. However, the configuration is the same for previous versions; only the screen names have changed.

Configuring 'RefID1' SP Adapter
[Help](#) | [Support](#) | [About](#) | [Logout \(Administrator\)](#)

[Main](#)
[Manage SP Adapter Instances](#)
[Create Adapter Instance](#)

Type | * **Instance Configuration** | [Actions](#) | [Extended 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.

The ReferenceID Adapter allows user attributes to be passed in and out of the PingFederate server via direct HTTP(S) calls. Attributes are retrieved via a ReferenceID.

Field Name	Field Value	Description
User Name	<input type="text"/> *	ID the application uses to authenticate with the PingFederate server.
Pass Phrase	<input type="text"/> *	Pass phrase the application uses to authenticate to the PingFederate server.
Allowed Subject DN	<input type="text"/>	Subject DN from the client certificate. If entered, PingFederate restricts client-certificate authentication (when enabled) by matching against this DN.
Allowed Issuer DN	<input type="text"/>	Issuer DN from the client certificate. If entered, PingFederate restricts client-certificate authentication (when enabled) by matching against this DN.
Logout Service Endpoint	<input type="text"/>	Application endpoint URL used for single logout.
Account Linking Authentication Endpoint	<input type="text"/>	The application endpoint URL where end users are redirected to obtain their local user IDs.

Show Advanced Fields

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

Field Name	Description
User Name	Enter an ID for the application to use when retrieving referenced attributes from PingFederate.
Pass Phrase	Use the next screen to display the clear-text value of the pass phrase you enter here, for copying to the application.
Allowed Subject DN	<p>(Optional) Client-certificate authentication may be used in addition to Basic authentication (User Name and Pass Phrase). To enable client-certificate-authentication, specify the Subject DN of the client certificate. Both Subject DN and Issuer DN must be specified to enable client-certificate authentication.</p> <p>Note: For information about configuring PingFederate to use this form of authentication in certain cases, see “Using Mutual SSL/TLS Authentication” on page 14.</p>
Allowed Issuer DN	(Optional) To enable client-certificate authentication, specify the Issuer DN of the incoming client certificate. Both Subject DN and Issuer DN must be specified to enable client certificate authentication. (See the Note above.)

Field Name	Description
Logout Service Endpoint	(Optional) Enter the SP-application URL where the user can initiate SAML single logout (SLO). SLO allows a user to log out of both the IdP and the SP sites with one action (for more information, see “Supported Standards” in the PingFederate manual <i>Getting Started</i>). For more information, see Logout Mode in the table for Advanced Fields under the next step.
Account Linking Authentication Endpoint	(Optional) Enter the SP-application URL where incoming SSO users can access IDs for local accounts, via SAML account linking. (For information about account linking, see the “Key Concepts” chapter of the PingFederate <i>Administrator’s Manual</i> .)

6. (Optional) Click **Show Advanced Fields** to view additional configuration settings.

You can change default values or settings, depending on your network configuration and other requirements at your site:

Field Name	Description
Reference Duration	PingFederate caches the reference and attributes for this amount of time. This field is provided for administrators to make adjustments, as needed, to address network latency issues.
Reference Length	Increasing the length of the reference makes it more difficult to replicate when security is a concern.
Require SSL/TLS	(Optional) We recommend using the secure transport protocol unless a secure, dedicated network segment exists between the application server and PingFederate.
Outgoing Attribute Format	As an option, you can change the format in which PingFederate encodes attribute values on the HTTP response to the application (see “ Attribute Formatting ” on page 15).
Incoming Attribute Format	As an option, you can change the expected format in which the application decodes attribute values on the HTTP request to PingFederate (see “ Attribute Formatting ” on page 15).
Logout Mode	Use these options to define how to handle application logout. Front Channel (the default) redirects the browser to the application endpoint, including the reference as a query parameter. When resolved this reference gives all of the user attributes as well as the resume path for the application to use in a logout response. Back Channel sends a direct HTTP request from the server to the application. The variable <code>\${attribute-name}</code> may be used for any attribute to build a dynamic URL.
Skip Host Name Validation	(Optional) Select the check box to skip host name validation, for example, when testing or when the host name validation cannot be performed.

7. Click **Next**.

8. (Optional) On the Actions screen, click **Show Pass Phrase**.

Use this option to copy and paste the pass phrase into the application to facilitate HTTP Basic authentication between the application and the PingFederate server.

9. Click **Next**.

10. (Optional) On the Extended Contract screen, add attributes you expect to retrieve in addition to the SAML subject (user ID).

(For more information about this screen, see the *PingFederate Administrator's Manual* or click **Help**.)

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

12. On the Manage SP Adapter Instances screen, click **Save** to complete the Adapter configuration.

13. Configure or modify the connection(s) to your IdP partner(s) using the ReferenceID Adapter Instance.

(For more information, see the "Service Provider SSO Configuration" chapter in the *PingFederate Administrator's Manual*.)

Using Mutual SSL/TLS Authentication

In addition to Basic authentication, applications may use client-certificate authentication to communicate with PingFederate and the ReferenceID Adapter. To use this authentication for PingFederate 6.2 and higher, the secondary SSL port must be configured, and application calls must use this port.

Your server may already be configured to use the secondary port for other back-channel SSO scenarios (for example, using SOAP). If not, follow this procedure:

1. 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.
2. Ensure that the port is configured in the Jetty container for client-certificate authentication:

In the directory:

`<pf-install>/pingfederate/server/default/deploy/jetty.sar/META-INF`

open the file `jboss-service.xml` and find the following lines:

```
<!-- - - - - - -->
<!-- Add a second HTTPS/SSL Connector -->
<!-- - - - - - -->
<!-- -->
<Call name="addConnector">
  <Arg>
    <New class="com.pingidentity.appserver.jetty.DynamicSslSocketConnector" >
      <Set name="Port"><SystemProperty name="pf.secondary.https.port" default="8443"/></Set>
      <Set name="Host"><SystemProperty name="pf.engine.bind.address"
        default="0.0.0.0"/></Set>
      <Set name="maxIdleTime">30000</Set>
      <Set name="NeedClientAuth">false</Set>
      <Set name="WantClientAuth">true</Set>
    </New>
  </Arg>
</Call>
```

Ensure that the value of `WantClientAuth` is set to `true`.

Application Integration

This section provides information developers need to integrate applications with PingFederate and the ReferenceID Adapter.

Authenticating to PingFederate

Either IdP or SP applications must authenticate to PingFederate via HTTP Basic authentication when making direct HTTP calls to drop off or pick up attributes. In addition, the application may present a trusted client certificate to PingFederate (see “[Using Mutual SSL/TLS Authentication](#)” on page 14).

Note: To simplify integration on platforms that do not provide native base-64 encoding support, applications may supply PingFederate with the user name and pass phrase for authentication via special HTTP headers named `ping.username` and `ping.pwd`, respectively.

Application Endpoints

The PingFederate URL an application uses for dropping off attributes is:

```
http[s]://<pf-host>:<pf-port>/ext/ref/dropoff
```

The PingFederate URL an application uses for picking up attributes is:

```
http[s]://<pf-host>:<pf-port>/ext/ref/pickup
```

(For more information, see the code snippets under “[Sample Java Code](#)” on page 16.)

Using HTTPS

Due to the sensitive nature of the authentication information and user attributes, SSL/TLS should always be used for communication between the application and PingFederate unless a secure and dedicated network segment exists between them.

SSL/TLS is the default transport setting for the ReferenceID Adapter configuration.

Attribute Formatting

Attribute formatting options are provided in the Advanced Fields section of the adapter configuration. By default PingFederate formats outgoing attributes and parses incoming attributes using JSON, a standard data structure for sending attributes in HTTP requests and responses.

For Outgoing Attributes

Outgoing attribute formatting is the process by which attribute names and values are encoded onto the HTTP response. PingFederate formats outgoing attributes at the pickup endpoint. How attributes are formatted affects the way your application parses them from the HTTP response.

As an alternative to JSON, Properties attribute formatting is provided, allowing applications to use the built in capabilities of the Java Properties class to parse the response.

For Incoming Attributes

Incoming attribute formatting is the process by which attribute names and values are decoded from the HTTP request. PingFederate parses incoming attributes at the drop-off endpoint. How attributes are parsed affects the way your application encodes them onto the HTTP request.

As an alternative to JSON, Query Parameter attribute formatting is provided, which allows applications to pass attribute names and values as request parameters on the query string, which must be URL encoded.

Sample Java Code

The following sections provide code snippets that may be modified for integration into Java applications.

Note: Basic client-side HTTP support is all that is required for non-Java applications to integrate with PingFederate using the ReferenceID Adapter.

IdP Application Code Snippet

The following code snippet provides an example IdP implementation using Java.

```
import javax.net.ssl.SSLContext;
import javax.net.ssl.TrustManager;
import javax.net.ssl.X509TrustManager;
import javax.net.ssl.SSLSocketFactory;
import javax.net.ssl.HttpsURLConnection;
import java.net.URL;
import java.net.URLConnection;
import java.security.cert.X509Certificate;
import java.security.cert.CertificateException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.io.IOException;
import java.io.OutputStreamWriter;

// This example uses JSON simple java toolkit available at
// http://code.google.com/p/json-simple/
import org.json.simple.JSONObject;
import org.json.simple.JSONValue;
import org.json.simple.parser.JSONParser;
import org.json.simple.parser.ContainerFactory;
import org.json.simple.parser.ParseException;

public class IdPSample
{
    public static void main(String[] args)
        throws Exception
```



```

{
    // Create a dummy X509TrustManager that will trust any server certificate
    // This is for example only and should not be used in production
    X509TrustManager tm = new X509TrustManager()
    {
        public void checkClientTrusted(X509Certificate[] x509Certs, String s)
            throws CertificateException
        {

        }

        public void checkServerTrusted(X509Certificate[] x509Certs, String s)
            throws CertificateException
        {

        }

        public X509Certificate[] getAcceptedIssuers()
        {
            return new X509Certificate[0];
        }
    };

    // Use the trust manager to get an SSLSocketFactory
    SSLContext sslContext = SSLContext.getInstance("TLS");
    sslContext.init(null, new TrustManager[] {tm}, null);
    SSLSocketFactory socketFactory = sslContext.getSocketFactory();

    // Create a JSON Object containing user attributes
    JSONObject idpUserAttributes = new JSONObject();
    idpUserAttributes.put("attribute1", "value1");
    idpUserAttributes.put("attribute2", "value2");
    idpUserAttributes.put("foo", "bar");

    // Drop the attributes into PingFederate
    String dropoffLocation = "https://localhost:9031/ext/ref/dropoff";
    System.out.println(dropoffLocation);
    URL dropUrl = new URL(dropoffLocation);
    URLConnection urlConnection = dropUrl.openConnection();
    HttpsURLConnection httpsURLConnection = (HttpsURLConnection)urlConnection;
    httpsURLConnection.setSSLSocketFactory(socketFactory);
    urlConnection.setRequestProperty("ping.uname", "changeme");
    urlConnection.setRequestProperty("ping.pwd", "this is a default example and
    should not be used in production");
    urlConnection.setRequestProperty("ping.instanceId", "idpadapter");

```

```

        // Write the attributes in URL Connection, this example uses UTF-8 encoding
        urlConnection.setDoOutput(true);
        OutputStreamWriter outputStreamWriter = new
        OutputStreamWriter(urlConnection.getOutputStream(), "UTF-8");
        idpUserAttributes.writeJSONString(outputStreamWriter);
        outputStreamWriter.flush();
        outputStreamWriter.close();

        // Get the response and parse it into a JSON object
        InputStream is = urlConnection.getInputStream();
        InputStreamReader streamReader = new InputStreamReader(is, "UTF-8");

        JSONParser parser = new JSONParser();
        JSONObject jsonRespObj = (JSONObject)parser.parse(streamReader);

        // Grab the value of the reference Id from the JSON Object. This value
        // must be passed to PingFederate on resumePath as the parameter 'REF'
        String referenceValue = (String)jsonRespObj.get("REF");
        System.out.println("Reference ID = " + referenceValue);

    }

    // Returns the json string by consuming the JSON input stream.
    protected static String getJSONString(InputStream jsonInputStream)
    throws IOException
    {
        String jsonString = null;
        char temp = (char)jsonInputStream.read();
        jsonString = "";
        while ((int)temp != 65535)
        {
            jsonString += temp;
            temp = (char)jsonInputStream.read();
        }
        return jsonString;
    }
}

```

SP Application Code Snippet

The following code snippet provides an example SP implementation using Java.

```

import javax.net.ssl.SSLContext;
import javax.net.ssl.TrustManager;
import javax.net.ssl.X509TrustManager;
import javax.net.ssl.SSLSocketFactory;
import javax.net.ssl.HttpURLConnection;
import java.net.URL;

```

```

import java.net.URLConnection;
import java.security.cert.X509Certificate;
import java.security.cert.CertificateException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.IOException;
import java.io.OutputStreamWriter;

// This example uses JSON simple java toolkit available at
// http://code.google.com/p/json-simple/
import org.json.simple.JSONObject;
import org.json.simple.JSONValue;
import org.json.simple.parser.JSONParser;

public class SPSample
{
    public static void main(String[] args)
        throws Exception
    {
        // Create a dummy X509TrustManager that will trust any server certificate
        // This is for example only and should not be used in production
        X509TrustManager tm = new X509TrustManager()
        {
            public void checkClientTrusted(X509Certificate[] x509Certs, String s)
                throws CertificateException
            {

            }

            public void checkServerTrusted(X509Certificate[] x509Certs, String s)
                throws CertificateException
            {

            }

            public X509Certificate[] getAcceptedIssuers()
            {
                return new X509Certificate[0];
            }
        };

        // Use the trust manager to get an SSLSocketFactory
        SSLContext sslContext = SSLContext.getInstance("TLS");
        sslContext.init(null, new TrustManager[] {tm}, null);
        SSLSocketFactory socketFactory = sslContext.getSocketFactory();

```

```

// Grab the value of the reference Id from the request, this
// will be sent by PingFederate as a query parameter 'REF'
String referenceValue = "<MUST_BE_REPLACED_BY_ACTUAL_VALUE>";

// Call back to PF to get the attributes associated with the reference
String pickupLocation = "https://localhost:9031/ext/ref/pickup?REF=" +
referenceValue;
System.out.println(pickupLocation);
URL pickUrl = new URL(pickupLocation);
URLConnection urlConn = pickUrl.openConnection();
HttpsURLConnection httpsURLConnection = (HttpsURLConnection)urlConn;
httpsURLConnection.setSSLSocketFactory(socketFactory);
urlConn.setRequestProperty("ping.uname", "changeme");
urlConn.setRequestProperty("ping.pwd", "please change me before you go into
production");
urlConn.setRequestProperty("ping.instanceId", "spadapter");

// Get the response and parse it into another JSON object which are the
// 'user attributes'.
// This example uses UTF-8 if encoding is not found in request.
String encoding = urlConn.getContentEncoding();
InputStream is = urlConn.getInputStream();
InputStreamReader streamReader = new InputStreamReader(is, encoding != null
? encoding : "UTF-8");

JSONParser parser = new JSONParser();
JSONObject spUserAttributes = (JSONObject)parser.parse(streamReader);
System.out.println("User Attributes received = " + spUserAttributes.toString());
}

}

```