

PingFederate®

Advanced SQL Datastore v1.3.2.0

User Guide



© 2005-2017 Ping Identity ® Corporation. All rights reserved.

PingFederate Advanced SQL Datastore User Guide
Version 1.3.2.0
March 2014 (Revised October 2017)

Ping Identity Corporation
1001 17th Street, Suite 100
Denver, CO 80202
USA

Phone: 877.898.2905 (+1 303.468.2882 outside North America)

Fax: 303.468.2909 E-mail: info@pingidentity.com

Web Site: <http://www.pingidentity.com>

Trademarks

Ping Identity, the Ping Identity logo, PingFederate, PingID, PingOne, PingConnect, and PingEnable are registered trademarks of Ping Identity Corporation (“Ping Identity”). All other trademarks or registered trademarks are the property of their respective owners.

Disclaimer

This document is proprietary and not for general publication. It may be provided ad hoc for informational purposes only, and the information herein is subject to change without notice. Ping Identity does not provide any warranties and specifically disclaims any liability in connection with this document.

Note that Ping Identity may not provide support for any sample configurations provided in this document. The variability inherent among security environments prevents full testing and support for all possible platform configurations. If you need special assistance or would like to inquire about implementation or support programs, please contact Ping Identity Global Client Services (<http://support.pingidentity.com>).

Table of Contents

Purpose	4
Prerequisites	4
Installation.....	4
Configuration	4
Testing	11
Logging.....	14

Purpose

This user guide is intended for use by PingFederate clients, who would like the ability to make stored procedure calls to a SQL database.

The Advanced SQL Datastore can be configured to call a stored procedure in order to retrieve identity data for the client application.

Prerequisites

This document assumes that you already have the following installed and configured:

- A functional PingFederate environment, version 8.4+
- JDK version 8+
- A pre-configured SQL datastore with driver (see <https://documentation.pingidentity.com/pingfederate/pf84/index.shtml#adminGuide/concept/configuringJdbcDatabaseConnection.html>)
- At least one IdP adapter for use as the primary form of authentication (e.g., HTML Form Adapter leveraging an LDAP Password Credential Validator (PCV))
- At least one SP connection that is connected with the primary form of authentication

Installation

1. From the /dist folder in *pf-advanced-sql-datastore-1.3.2.0.zip*, copy the noted file to the following directory in your PingFederate:
 - <PingFederateInstall>/pingfederate/server/default/deploy/
 - pf-advanced-sql-datastore-1.3.2.0.jar
 - commons-dbutils-1.7.jar
2. Repeat step 1 on other clustered engine nodes.
3. Start or restart PingFederate.

Configuration

Configuring the Custom Datastore

Please note: this example is using Microsoft SQL Server for the SQL database.

1. Log into the PingFederate admin console and click **Data Stores** under **Server Configuration >> System Settings**.
2. Ensure that a pre-configured SQL datastore has already been configured. If not, please follow the instructions at:
https://documentation.pingidentity.com/pingfederate/pf84/index.shtml - concept_configuringJdbcDatabaseConnection.html
3. Click **Add New Data Store...**
4. Choose **Custom**, and click **Next**.

PingFederate

MAIN

- IDP Configuration
- SP Configuration
- OAuth Settings
- Server Configuration**

Manage Data Stores | Data Store

Data Store Type | Custom Data Store Type | Configure Attribute Source Adapter Instance | Summary

Please select a type of data store.

☐ DATABASE
☐ LDAP
☒ CUSTOM

Cancel Next

- Enter the **Data Store Instance Name**, choose **Advanced SQL Datastore 1.3.2.0** as the **Data Store Type**, and click **Next**.

PingFederate

MAIN

- IDP Configuration
- SP Configuration
- OAuth Settings
- Server Configuration**

Manage Data Stores | Data Store

Custom Data Store Type | Configure Attribute Source Adapter Instance | Summary

Enter a descriptive name, a system-wide unique ID, and select the custom data store adapter to use.

DATA STORE INSTANCE NAME: AdvancedSQLDatastore

DATA STORE TYPE: Advanced SQL Datastore 1.3.2.0

☐ MASK VALUES IN LOG

Cancel Next Done Save

- If the stored procedure requires input parameter(s), click **Add a new row to 'Stored Procedure Input Parameters'**, add accordingly, and to the order in which they are to be inputted into the stored procedure call. Click the **Update** link to complete the addition(s).

Manage Data Stores | Data Store

Custom Data Store Type	Configure Attribute Source Adapter Instance	Summary
------------------------	---	---------

Configure the Custom Source Adapter.

Advanced SQL Datastore 1.3.2.0

STORED PROCEDURE INPUT PARAMETER

(Please add a row for each required input parameter in the order expected by the stored procedure.)

INPUT PARAMETER NAME (The name of the input parameter column.)	INPUT PARAMETER DATA TYPE (The input parameter data type.)	Action		
@subjectParameter1	STRING	Move down	Edit	Delete
@subjectParameter2	INTEGER	Move up Move down	Edit	Delete
@subjectParameter3	DATE	Move up	Edit	Delete

[Add a new row to 'Stored Procedure Input Parameter'](#)

- Click **Add a new row to 'Stored Procedure Output Parameters'** and add the identity attribute(s) to be returned from the stored procedure call. Click the **Update** link to complete the addition(s).

STORED PROCEDURE OUTPUT PARAMETER

(Please add a row for each output parameter that the stored procedure will return.)

OUTPUT PARAMETER NAME (The name of the output parameter column.)	OUTPUT PARAMETER DATA TYPE (The output parameter data type.)	Action		
Claim1	STRING	Move down	Edit	Delete
Claim2	INTEGER	Move up Move down	Edit	Delete
Claim3	DATE	Move up	Edit	Delete

[Add a new row to 'Stored Procedure Output Parameter'](#)

- From the **Database Data Store** dropdown, select the pre-configured SQL datastore, enter the **Stored Procedure Name**, and click **Next**.

Field Name	Field Value	Description
DATABASE DATA STORE	<code>jdbc:sqlserver://mssql-lab.pingidentity.com</code> ▼	Please choose the proper JDBC data store.
STORED PROCEDURE NAME	<code>dbo.GetUserIdentityData</code>	Enter the name of the stored procedure to be executed (e.g., MS SQL: <code>dbo.StoredProcedureName</code> ; Oracle/MySQL: <code>storedProcedureName</code>).

9. Review the configuration on the summary page, and click **Done**.

PingFederate

MAIN

- IdP Configuration
- SP Configuration
- OAuth Settings
- Server Configuration**

Manage Data Stores | Data Store

Custom Data Store Type
Configure Attribute Source Adapter Instance
Summary

Click a heading link to edit a configuration setting.

Data Store

Custom Data Store Type

Data Store Instance Name	AdvancedSQLDatastore
Data Store Type	Advanced SQL Datastore 1.3.2.0
Attribute Source Adapter Class Name	com.pingidentity.clientservices.product.datastore.sql.AdvancedSQLDatastore

Configure Attribute Source Adapter Instance

Stored Procedure Input Parameter	@subjectParameter1, STRING
Stored Procedure Input Parameter	@subjectParameter2, INTEGER
Stored Procedure Input Parameter	@subjectParameter3, DATE
Stored Procedure Output Parameter	Claim1, STRING
Stored Procedure Output Parameter	Claim2, INTEGER
Stored Procedure Output Parameter	Claim3, DATE
Database Data Store	jdbc:sqlserver://mssql-lab.pingidentity.com\\SQLEXPRESS:1433;databaseName=Users
Stored Procedure Name	dbo.GetUserIdentityData

Cancel
Previous
Done
Save

Configuring the SP Connection to Leverage the Custom Datastore

1. Click the SP connection to be modified, which should be located under **IdP Configuration >> SP Connections**.
2. Under **Assertion Creation**, click on **Attribute Sources & User Lookup**, and click **Add Attribute Source**.

3. Enter the **Attribute Source Description**, select the **Active Data Store** that was configured in the **Configuring the Custom Datastore** section above, and click **Next**.

PingFederate

MAIN

IDP IDP Configuration

SP SP Configuration

OA OAuth Settings

Server Configuration

SP Connections | SP Connection | Browser SSO | Assertion Creation | IdP Adapter Mapping | Attribute Sources & User Lookup

Data Store | Configure Custom Source Filters | Configure Custom Source Fields | 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 ID AdvSQLDS

ATTRIBUTE SOURCE DESCRIPTION AdvSQLDS

ACTIVE DATA STORE AdvancedSQLDatastore

DATA STORE TYPE Custom

Manage Data Stores

Cancel Next

4. Enter the **Stored Procedure Input Parameter Data**. Follow the instructions that are summarized in the description for the correct format. Click **Next**.

PingFederate

MAIN

IDP IDP Configuration

SP SP Configuration

OA OAuth Settings

Server Configuration

SP Connections | SP Connection | Browser SSO | Assertion Creation | IdP Adapter Mapping | Attribute Sources & User Lookup

Data Store | Configure Custom Source Filters | Configure Custom Source Fields | Summary

Perform any filter configuration for the Custom Source.

Field Name	Field Value	Description
STORED PROCEDURE INPUT PARAMETER DATA	<code>\${username} 999 12-31-2015:11:59:59</code>	Enter the input parameter value(s) in the order expected by the stored procedure (adapter identity attribute(s) should be in the proper syntax, e.g., <code>\${username}</code>). Each input parameter value should be separated by a pipe (<code> </code>). Make sure there are no spaces before and after the pipe. If the value is null, either input <code>'null'</code> or leave it empty. An example: <code>\${username} Text1, \${role} Text2 100 01-01-2014 00:01:01 null Text3 - Text4</code>

Cancel Previous Next

5. Check the identity attributes to be returned (these are the output parameters returned from the stored procedure call that was configured in step 6 under the **Configure the Custom Datastore** section), and click **Next**.

PingFederate

MAIN

IDP Configuration

SP Configuration

OAuth Settings

Server Configuration

SP Connections | SP Connection | Browser SSO | Assertion Creation | IdP Adapter Mapping | Attribute Sources & User Lookup

Data Store | Configure Custom Source Filters | Configure Custom Source Fields | Summary

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

☒ CLAIM1
☒ CLAIM2
☒ CLAIM3

Cancel Previous Next

6. Review **Summary**. Then click **Done**.

PingFederate

MAIN

IDP Configuration

SP Configuration

OAuth Settings

Server Configuration

SP Connections | SP Connection | Browser SSO | Assertion Creation | IdP Adapter Mapping | Attribute Sources & User Lookup

Data Store | Configure Custom Source Filters | Configure Custom Source Fields | Summary

Attribute Source Summary

Attribute Sources & User Lookup

Data Store

Attribute Source	AdvSQLDS
Attribute Source Id	AdvSQLDS
Type of Data Store	Custom
Data Store	AdvancedSQLDatastore

Configure Custom Source Filters

Stored Procedure Input Parameter Data	\${username}999/12-31-2015:11:59:59
---------------------------------------	-------------------------------------

Configure Custom Source Fields

Field	Claim1
Field	Claim2
Field	Claim3

Cancel Previous Done Save

7. Click **Next**.

PingFederate

MAIN

- IDP Configuration
- SP Configuration
- OAuth Settings
- Server Configuration

SP Connection | Browser SSO | Assertion Creation | IdP Adapter Mapping

Adapter Instance | Mapping Method | Attribute Sources & User Lookup | Attribute Contract Fulfillment | Issuance Criteria | Summary

Here you can specify a series of local data stores that will be used to supply additional information about the user in the SAML assertion to the SP.

Description	Type	Action
AdvSQLDS	Custom	Delete

Add Attribute Source

Cancel
Previous
Next
Done
Save

8. Map the **Attribute Contract** accordingly, and click **Next**.

PingFederate

MAIN

- IDP Configuration
- SP Configuration
- OAuth Settings
- Server Configuration

SP Connections | SP Connection | Browser SSO | Assertion Creation | IdP Adapter Mapping

Adapter Instance | Mapping Method | Attribute Sources & User Lookup | Attribute Contract Fulfillment | Issuance Criteria | Summary

Fulfill your Attribute Contract with values from one or more data stores, the authentication adapter, or dynamic text values.

Attribute Contract	Source	Value	Actions
Attribute1	Custom (AdvSQLDS)	Claim1	None available
Attribute2	Custom (AdvSQLDS)	Claim2	None available
Attribute3	Custom (AdvSQLDS)	Claim3	None available
SAML_SUBJECT	Adapter	username	None available

Cancel
Previous
Next
Done
Save

9. Click **Next**.

10. Review the configuration on the summary page, and click **Done**.

PingFederate

MAIN

IDP IDP Configuration

SP SP Configuration

OA OAuth Settings

Server Configuration

Copyright © 2003-2015
Ping Identity Corporation
All rights reserved
Version 8.0.3.1

SP Connections | SP Connection | Browser SSO | Assertion Creation | IdP Adapter Mapping

Adapter Instance

Mapping Method

Attribute Sources & User Lookup

Attribute Contract Fulfillment

Issuance Criteria

Summary

Click a heading link to edit a configuration setting.

Adapter Instance

Selected adapterHTMLFormIdPAdapterAD

Mapping Method

AdapterHTML Form IdP Adapter

Mapping MethodRetrieve additional attributes from multiple data stores using one mapping

Attribute Sources & User Lookup

Data StoreMultiAdDs (Custom)

Data StoreAdvSQLDS (Custom)

Attribute Sources & User Lookup

Data Store

Attribute SourceAdvSQLDS

Attribute Source IdAdvSQLDS

Type of Data StoreCustom

Data StoreAdvancedSQLDatastore

Configure Custom Source Filters

Stored Procedure Input Parameter Data\${username}999112-31-15 11:59:59

Configure Custom Source Fields

FieldClaim1

FieldClaim2

FieldClaim3

Attribute Contract Fulfillment

SAML_SUBJECTusername (Adapter)

Attribute1Claim1 (Custom)

Attribute3Claim3 (Custom)

Attribute2Claim2 (Custom)

Issuance Criteria

Criterion(None)

11. Click **Done**.
12. Click **Done**.
13. Click **Done**.
14. Make sure the connection is **Active**. Click **Save**.
15. Click **Save**.

Testing

Primary Test Case

1. Open a browser and go to the IdP login form chosen as the primary form of authentication. In this example, the HTML Form Adapter leveraging an LDAP PCV was chosen.

11


Sign On

Please sign on and we'll send you right along.

Username

Password

☐ Remember my username


Cancel
Sign On

- Log in using the test credentials.
- If authorized into the target destination, authentication was a success.

This PingFederate log snippet showing a successful SAML transaction, which contains data pulled by the configured Advanced SQL Datastore via a stored procedure call:

```

11:42:00,088 DEBUG [LoggingInterceptor] Received InMessageContext:
InMessageContext
XML: <samlp:Response Version="2.0" ID="tfLulfv1TRfU2rsgz-MpOaicIPB" IssueInstant="2014-05-14T18:41:57.962Z" xmlns:S11="http://schemas.xmlsoap.org/soap/envelope/" xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol">
  <saml:Issuer
xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">localhost:default:entityId</saml:Issuer>
  <samlp:Status>
    <samlp:StatusCode Value="urn:oasis:names:tc:SAML:2.0:status:Success"/>
  </samlp:Status>
  <saml:Assertion ID="ATjv3cl0j1C8jq87mvEKgkHFAB2" IssueInstant="2014-05-14T18:41:59.887Z" Version="2.0" xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">
    <saml:Issuer>localhost:default:entityId</saml:Issuer>
    <saml:Subject>
      <saml:NameID Format="urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified">user1@pfed-lab.com</saml:NameID>
      <saml:SubjectConfirmation Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">
        <saml:SubjectConfirmationData Recipient="https://localhost:9031/sp/ACS.saml2" NotOnOrAfter="2014-05-14T18:46:59.891Z"/>
      </saml:SubjectConfirmation>
    </saml:Subject>
  </saml:Assertion>
</samlp:Response>

```

```

<saml:Conditions NotBefore="2014-05-14T18:36:59.891Z" NotOnOrAfter="2014-05-
14T18:46:59.891Z">
  <saml:AudienceRestriction>
    <saml:Audience>localhost:default:entityId</saml:Audience>
  </saml:AudienceRestriction>
</saml:Conditions>
<saml:AuthnStatement SessionIndex="ATjv3cl0j1C8jq87mvEKgkHFAB2"
AuthnInstant="2014-05-14T18:41:59.882Z">
  <saml:AuthnContext>
    <saml:AuthnContextClassRef>urn:oasis:names:tc:SAML:2.0:ac:classes:unspecified</saml:
AuthnContextClassRef>
  </saml:AuthnContext>
</saml:AuthnStatement>
<saml:AttributeStatement>
  <saml:Attribute Name="Claim1" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-
format:basic">
    <saml:AttributeValue xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">user1@pfed-
lab.com</saml:AttributeValue>
    <saml:AttributeValue xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">999</saml:AttributeValue>
  </saml:Attribute>
  <saml:Attribute Name="Claim2" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-
format:basic">
    <saml:AttributeValue xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">user1@pfed-
lab.com</saml:AttributeValue>
    <saml:AttributeValue xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">999</saml:AttributeValue>
  </saml:Attribute>
  <saml:Attribute Name="Claim3" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-
format:basic">
    <saml:AttributeValue xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">12-31-15
11:59:59</saml:AttributeValue>
  </saml:Attribute>
</saml:AttributeStatement>
</saml:Assertion>
</samlp:Response>
entityId: localhost:default:entityId (IDP)
Binding: urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact
params:
{SAMLart=AAQAAFqZ+kMVVykI6CFQoTzcH/ejLoCnv/6WUgxfiT+IOm7gr4genTzLScc=}
SignatureStatus: NOT_PRESENT
Binding says to sign: false

```

Other Test Cases

Please note: For all test cases below, please make sure to log out, clear browser data, close and re-open the browser.

1. Repeat the primary test case as defined above, but with a test user that should fail (i.e., an empty result would be returned by the stored procedure call).
2. Repeat the primary test case as defined above, but with a test user that would have multiple data returned as a result from the stored procedure call. (Note: the above sample PingFederate log data shows a multiple data return for the attributes, Claim1, Claim2, Claim3).

Logging

To enable various logging modes for the Advanced SQL Datastore, add the following in the logger section (<Loggers>...</Loggers>) to

<PingFederateInstall>/pingfederate/server/default/conf/log4j2.xml:

```
<Logger name="com.pingidentity.clientservices.product.datastore.sql" level="[ DEBUG |
INFO | WARN | ERROR ]" />
```

To enable logging for the Advanced SQL Datastore in a separate file, add the following in the relevant sections in <PingFederateInstall>/pingfederate/server/default/conf/log4j2.xml.

```
<RollingFile name="AdvancedSQLDS" fileName="${sys:pf.log.dir}/advancedsqllds.log"
filePattern="${sys:pf.log.dir}/advancedsqllds.%d{yyyy-MM-dd}.log"
ignoreExceptions="false">
  <PatternLayout>
    <!-- Uncomment this if you want to use UTF-8 encoding instead of system's
    default encoding.
    <charset>UTF-8</charset> -->
    <pattern>%d %m%n</pattern>
  </PatternLayout>
  <Policies>
    <TimeBasedTriggeringPolicy />
  </Policies>
</RollingFile>

<Logger name="com.pingidentity.clientservices.product.datastore.sql"
level="[ DEBUG | INFO | WARN | ERROR ]" additivity="false" includeLocation="true"/>
  <appender-ref ref="AdvancedSQLDS" />
</Logger>
```

Detailed training on using Log4j in PingFederate can be found at:

<https://ping.force.com/Support/PingIdentityKnSearchHome?searchText=log4j>

This is an example of a PingFederate log snippet for the Advanced SQL Datastore that has the log level set for DEBUG:

```
13:29:34,301 DEBUG [AdvancedSQLDatastore] retrieveValues :: BEGIN
13:29:34,303 DEBUG [AdvancedSQLDatastore] retrieveValues :: storedProcedure :: dbo.
GetUserIdentityData (?, ?, ?, ?, ?, ?)
```

```

13:29:34,621 DEBUG [AdvancedSQLDatastore] retrieveValues :: ps - setting this to String :: jdoe
13:29:34,631 DEBUG [AdvancedSQLDatastore] retrieveValues :: ps - setting this to Integer:: 999
13:29:34,631 DEBUG [AdvancedSQLDatastore] retrieveValues :: ps - setting this to Date:: 12-31-
2013 11:59:59
13:29:34,763 DEBUG [AdvancedSQLDatastore] retrieveValues :: response time (ms) for calling
stored procedure :: 131.852
13:29:34,791 DEBUG [AdvancedSQLDatastore] retrieveValues :: Returning out.getName() ::
Claim1
13:29:34,792 DEBUG [AdvancedSQLDatastore] retrieveValues :: Returning out.getValue() ::
[[Claim1 Text1, Claim1 Text2]
13:29:34,844 DEBUG [AdvancedSQLDatastore] retrieveValues :: END
13:29:34,844 DEBUG [CustomAttributeSource] Custom Attr Src search result: {Claim1=[Claim1
Text1, Claim1 Text2]}
13:29:34,845 DEBUG [AttributeMapping] Source
attributes:{context.AuthenticationCtx=urn:oasis:names:tc:SAML:2.0:ac:classes:unspecified,
context.ClientIp=0:0:0:0:0:0:1, context.VirtualServerId=localhost:default:entityId, username= jdoe,
context.HttpRequest=/idp/jLmFF/resumeSAML20/idp/startSSO.ping} Attributes from Datasource:{
AdvSqlIDs.Claim1=[Claim1 Text1, Claim1 Text2]} Resulting attributes:{SAML_SUBJECT=jdoe,
Claim1=[Claim1 Text1, Claim1 Text2]}

```