Ping Identity[™]

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/configuringIdbcDatabaseConnection.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

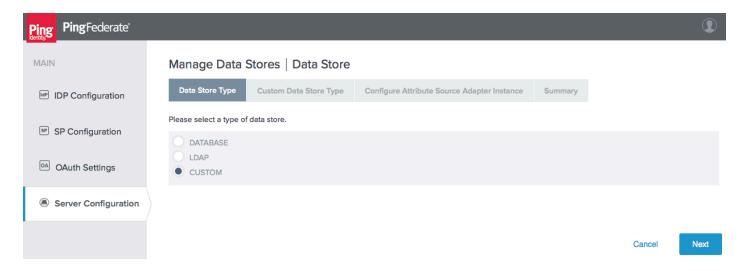
- 1. From the /dist folder in *pf-advanced-sql-datastore-1.3.2.0.zip*, copy the noted file to the following directory in your PingFederate:
 - <PingFederateInstall>/pingfederate/server/default/deploy/
 - o pf-advanced-sql-datastore-1.3.2.0.jar
 - o commons-dbutils-1.7.jar
- 2. Repeat step 1 on other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

Configuration

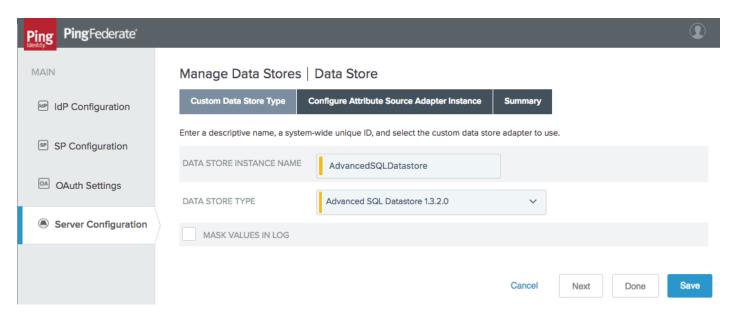
Configuring the Custom Datastore

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

- 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...
- Choose Custom, and click Next.

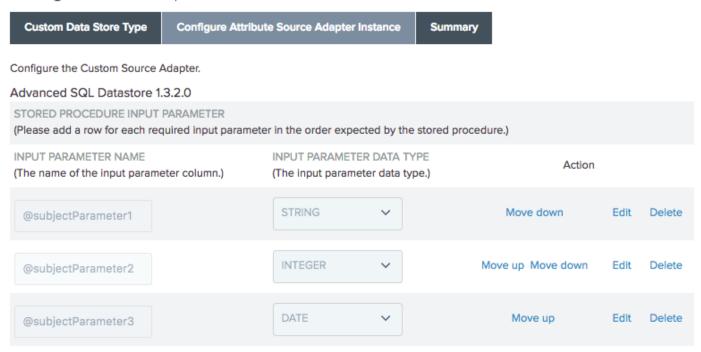


5. Enter the **Data Store Instance Name**, choose **Advanced SQL Datastore 1.3.2.0** as the **Data Store Type**, and click **Next**.



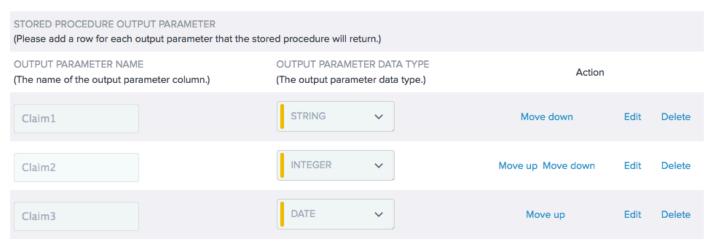
6. If the stored procedure requires input parameter(s), click **Add a new row to 'Stored Procedure Input Parameters'**, add accordingly, and to the order in which they are to be inputted into the stored procedure call. Click the **Update** link to complete the addition(s).

Manage Data Stores | Data Store



Add a new row to 'Stored Procedure Input Parameter'

7. 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).

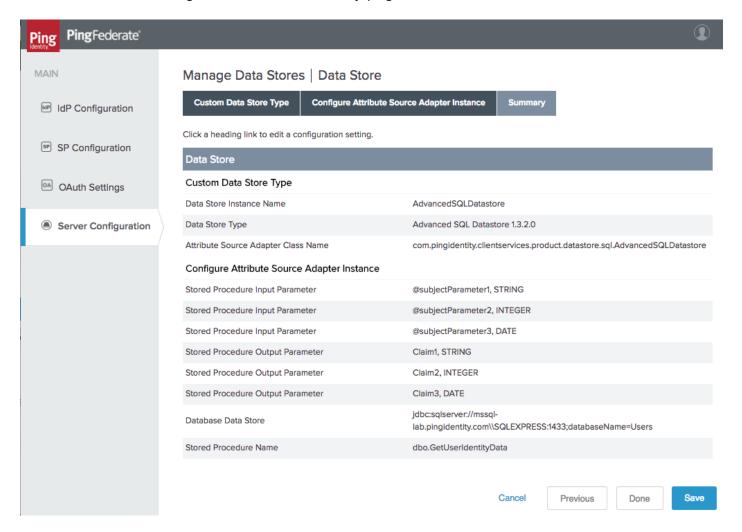


Add a new row to 'Stored Procedure Output Parameter'

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



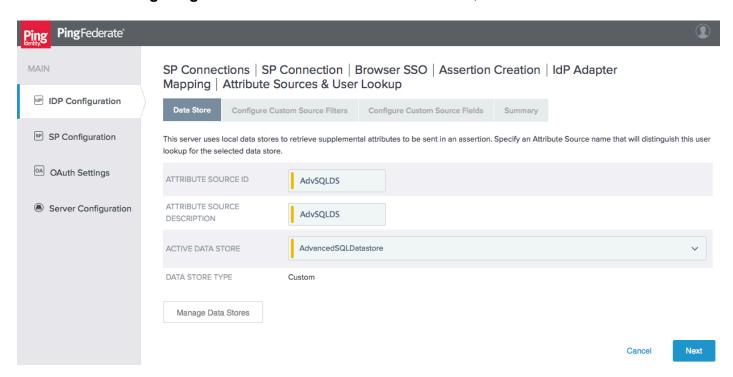
9. Review the configuration on the summary page, and click **Done**.



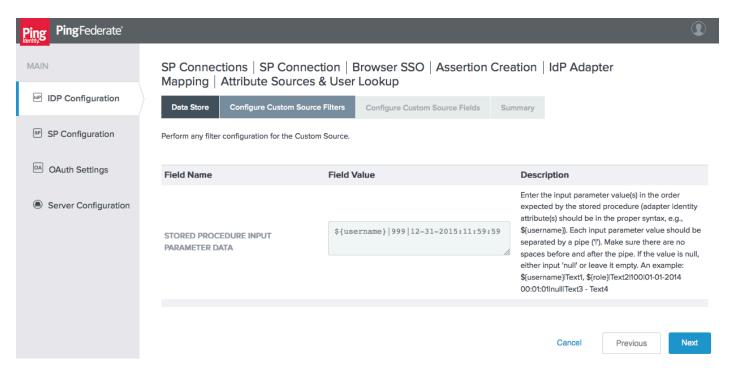
Configuring the SP Connection to Leverage the Custom Datastore

- 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.

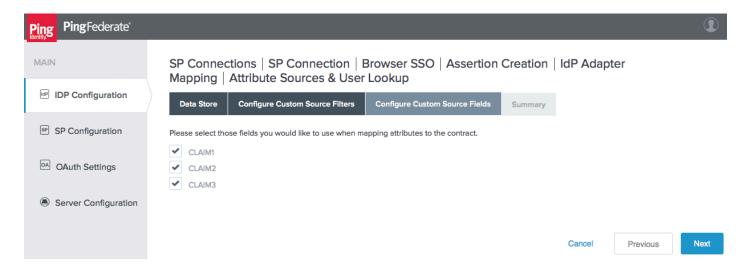
3. Enter the **Attribute Source Description**, select the **Active Data Store** that was configured in the **Configuring the Custom Datastore** section above, and click **Next**.



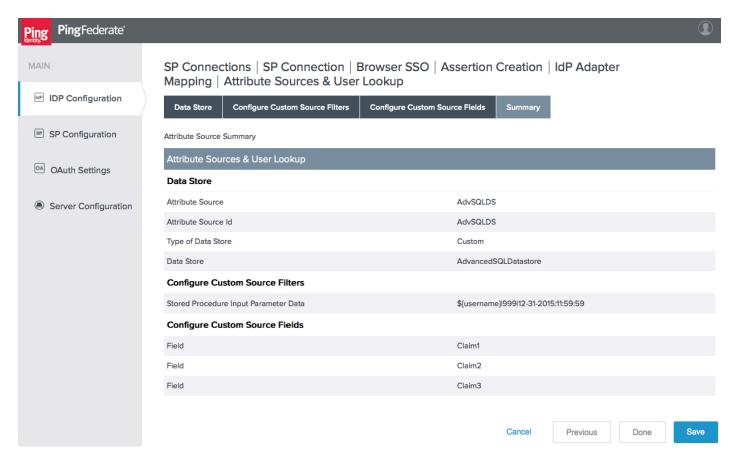
4. Enter the **Stored Procedure Input Parameter Data**. Follow the instructions that are summarized in the description for the correct format. Click **Next**.



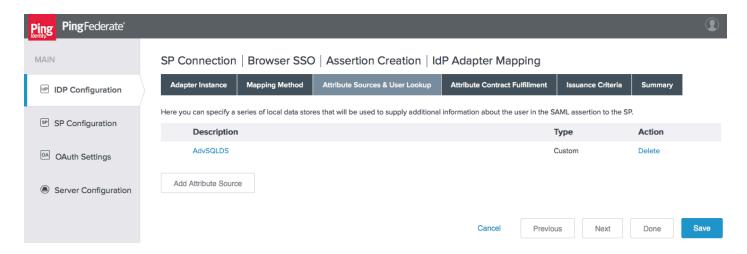
5. Check the identity attributes to be returned (these are the output parameters returned from the stored procedure call that was configured in step 6 under the *Configure the Custom Datastore* section), and click **Next**.



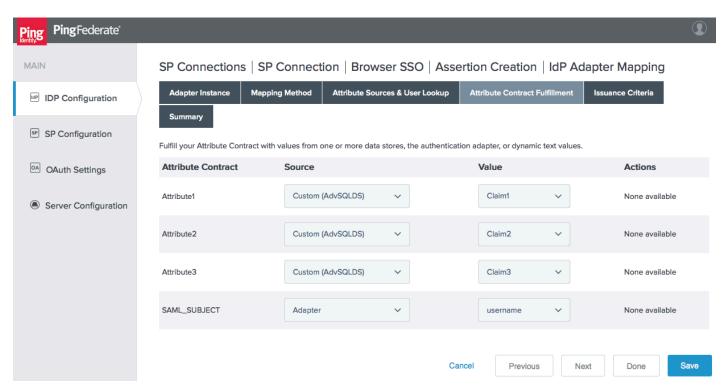
6. Review Summary. Then click Done.



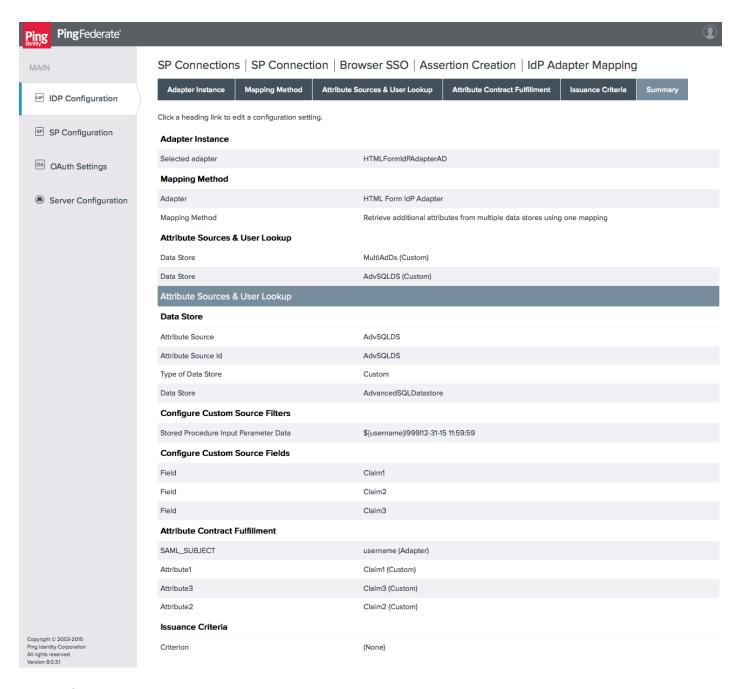
7. Click Next.



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



- 9. Click Next.
- 10. Review the configuration on the summary page, and click **Done**.



- 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.

Sign On		
Please sign on and we'll send you right along.		
Username		
Ukanggradigan		
Password		
PSe season of		
Remember my username		
Ping	Cancel	Sign On

- 2. Log in using the test credentials.
- 3. 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:lssuer

xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">localhost:default:entityId</saml:lssuer> <samlp:Status>

<samlp:StatusCode Value="urn:oasis:names:tc:SAML:2.0:status:Success"/>

</samlp:Status>

<saml:Assertion ID="ATjv3cl0j1C8jq87mvEKgkHFAB2" IssueInstant="2014-05-</p>

14T18:41:59.887Z" Version="2.0" xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"> <saml:lssuer>localhost:default:entityId</saml:lssuer>

<saml:Subject>

<saml:NameID Format="urn:oasis:names:tc:SAML:1.1:nameid-</p>

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"</p>

NotOnOrAfter="2014-05-14T18:46:59.891Z"/>

</saml:SubjectConfirmation>

</saml:Subject>

```
<saml:Conditions NotBefore="2014-05-14T18:36:59.891Z" NotOnOrAfter="2014-05-</p>
14T18:46:59.891Z">
   <saml:AudienceRestriction>
     <saml:Audience>localhost:default:entityId</saml:Audience>
   </saml:AudienceRestriction>
  </saml:Conditions>
  <saml:AuthnStatement SessionIndex="ATjv3cl0j1C8jq87mvEKgkHFAB2"</p>
AuthnInstant="2014-05-14T18:41:59.882Z">
   <saml:AuthnContext>
   <saml:AuthnContextClassRef>urn:oasis:names:tc:SAML:2.0:ac:classes:unspecified</saml:</p>
   AuthnContextClassRef>
   </saml:AuthnContext>
  </saml:AuthnStatement>
  <saml:AttributeStatement>
    <saml:Attribute Name="Claim1" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-</p>
format:basic">
    <saml:AttributeValue xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema"</p>
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"</p>
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-</p>
format:basic">
     <saml:AttributeValue xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema"</p>
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"</p>
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-</p>
format:basic">
     <saml:AttributeValue xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema"</p>
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+kMVVykl6CFQoTzcH/ejLoCnv/6WUgxfiT+IOm7gr4genTzLScc=}
SignatureStatus: NOT PRESENT
Binding says to sign: false
```

Other Test Cases

<u>Please note</u>: 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="${sys:pf.log.dir}/advancedsglds.log"</p>
filePattern="${sys:pf.log.dir}/advancedsqlds.%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"</p>
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/PingIdentityKnbSearchHome?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.Clientlp=0:0:0:0:0:0:0:0:1, context.VirtualServerId=localhost:default:entityId, username= jdoe, context.HttpRequest=/idp/jLmFF/resumeSAML20/idp/startSSO.ping} Attributes from Datasource:{ AdvSqlDs.Claim1=[Claim1 Text1, Claim1 Text2]} Resulting attributes:{SAML_SUBJECT=jdoe, Claim1=[Claim1 Text1, Claim1 Text2]}