Contents

Post patch deployment steps for Connect Server

In this release, some manual changes are required in Connect Server.

NOTE:

- *Ensure that the Kernel and Webrunner are not running before you perform the following changes in Connect Server.*
- *It is assumed that you have followed all the prerequisites before applying the Connect Server patch.*

Changes in web.xml

1. Copy the **web.xml** file from patch location, and paste it at **…<ConnectServerInstallFolder>/AdeptiaServer/ServerKernel/web/WEB-INF** location replacing the existing one.
2. Open the copied file and replicate the manual changes that you may have done in your previous **web.xml** file (that you had saved as a backup).
3. Save the file.

Changes in jetty.xml

1. Copy the **jetty.xml** file from patch location, and paste it at **…<ConnectServerInstallFolder>/AdeptiaServer/ServerKernel/etc/jetty** location replacing the existing one.
2. Open the copied file and replicate the manual changes that you may have done in your previous **jetty.xml** file (that you had saved as a backup).
3. Save the file.

Changes in webdefault.xml

1. Copy the **webdefault.xml** file from patch location, and paste it at **…<ConnectServerInstallFolder>/AdeptiaServer/ServerKernel/web/WEB-INF** location replacing the existing one.
2. Open the copied file and replicate the manual changes that you may have done in your previous **webdefault.xml** file (that you had saved as a backup).
3. Save the file.

Changes in launcher.properties

1. Copy the **launcher.properties_Server** file from patch location, and paste it at **…<ConnectServerInstallFolder>/AdeptiaServer/ServerKernel/etc** location.
2. Rename the copied **launcher.properties_Server** file to **launcher.properties**.
3. Open the copied file and replicate the manual changes that you may have done in your previous **launcher.properties** file (that you had saved as a backup).
4. Save the file.

Changes in server-configure.properties

In case you are using SQL server database, you need to add the parameters **encrypt=true** and **trustServerCertificate=true** to the JDBC URL for backend and log database in the relevant sections of server-configure.properties file. The JDBC URL for backend database can be found under "Database Configuration" section and that for the log database can be found under "Database Appender" section. Examples for both backend and log db JDBC URLs is given below:

Backend db JDBC URL:

jdbc:sqlserver://<IP address>:<Port>;DataBaseName=<Name of backend db>;encrypt=true;trustServerCertificate=true;

Log db JDBC URL:

jdbc:sqlserver://<IP address>:<Port>;DataBaseName=<Name of log db>;encrypt=true;trustServerCertificate=true;

Changes in quartz.properties

In case you are using SQL server database, you need to add the parameters **encrypt=true** and **trustServerCertificate=true** to the JDBC URL for backend database given for the property **org.quartz.dataSource.aBPM.URL** in quartz.properties file. Example for backend db JDBC URL is given below:

jdbc:sqlserver://<IP address>:<Port>;DataBaseName=<Name of backend db>;encrypt=true;trustServerCertificate=true;

Post patch deployment steps for Connect Portal

In this release, some manual changes are required in Connect Portal.

NOTE:

- *Ensure that the Portal is not running before you perform the following changes in Connect Portal.*
- *It is assumed that you have followed all the prerequisites before applying the Connect Server patch.*

Changes in launcher.properties

1. Copy the **launcher.properties_Portal** file from patch location, and paste it at **...<ConnectPortalInstallFolder>/ConnectPortal/Conf** location.
2. Rename the copied **launcher.properties_Portal** file to **launcher.properties**.
3. Open the copied file and replicate the manual changes that you may have done in your previous **launcher.properties** file (that you had saved as a backup).
4. Save the file.

Changes in server.xml

1. Copy the **server.xml** file from the patch location, and paste it at **...<ConnectPortalInstallFolder>/ConnectPortal/Conf** location replacing the existing one.
2. Open the copied file and replicate the manual changes that you may have done in your previous **server.xml** file (that you had saved as a backup).
3. Save the file.

After you have made all the manual changes, start the Kernel, Webrunner, and Portal.

Steps to deal with impacts of JRE upgrade

Adeptia Connect version 3.7.2 comes with an upgraded Amazon Corretto JRE. The bundled Amazon Corretto JRE in this version of Adeptia Connect has been upgraded from v8 to v17 to ensure improved security. This upgrade has the following impacts and the actions you must take to deal with them.

Changes in custom jars or custom classes

The upgrade to Java 17 may require you to handle the conflicts that may occur with custom jars or custom classes. You may come across the following scenarios wherein you need to take the necessary actions to address the conflicts.

1. Java packages used in the custom plugins, custom jars, or custom classes may have changed. To address this issue, you need to change the custom plugins, custom jars, and custom classes accordingly.
2. Classes used in custom plugins, custom jars, or custom classes may have been deprecated or their implementation may have changed. To address this issue, you need to change the custom plugins, custom jars, and custom classes accordingly.
3. Although Java 17 is backward compatible – code compiled with Java 8 continues to run on Java 17 most of the time – there are certain restrictions implemented in Java 17, for example, Java 17 restricts the use of certain packages and reflections. However, you can override these restrictions by adding the required entries in the launcher.properties file.
4. External jar(s) in the **ext** folder may not be compatible with the upgraded versions of the framework. In this case, jars need to be upgraded accordingly.

Changes in JMS Provider

The JMS Provider that connects to Apache Active MQ JMS server uses the two jars **activemq-core-5.7.0.jar** and **geronimo-j2ee-management_1.1_spec-1.0.1.jar.** Ensure that:

1. You use these two jars when you create a JMS provider that connects to Apache Active MQ JMS server.
2. You have updated any existing JMS provider that connects to Apache Active MQ JMS server to use these two jars.

Changes in SAML

The SAML implementation in Adeptia Connect has got configuration changes. The following sections contain updated information with respect to the changes.

Exposing IdP Metadata to Adeptia Connect/AIS

Exposing IdP metadata information to Adeptia Connect/AIS allows the Service Provider to read the details of IdP server from a particular location.

Depending upon the IdP server you are using (for example, ADFS, PingFederate), download the IdP server metadata file, rename it to **idp.xml,** and place it at the location **...<ConnectPortalInstallFolder>/resources_config/saml.**

After you have placed the idp.xml file, you need to uncomment the property **SAML_SSO_IDPS_CONFIGURATION_0_METADATA_LOCATION** in saml.properties file located at ...**<ConnectPortalInstallFolder>/resources_config/saml**.

Configuring multiple SAML IdPs (Valid only for AC v3.7.2)

This page helps you in exposing the metadata of multiple Identity Providers to Adeptia Connect/ AIS. Exposing IdP metadata information to Adeptia Connect/AIS allows Service Provider to read the details of IdP server from a particular location.

This section contains the following information.

Prerequisite

Before you start configuring multiple IdPs, ensure that you have met the following prerequisites:

- Depending upon the IdPs (for example, ADFS, PingFederate), download the respective IdP server metadata files.
- Rename the metadata files (for example, **idp1.xml, idp2.xml**).
- Place the files at the location **...<ConnectPortalInstallFolder>/resources_config/saml.**

  You can also place a metadata file at any other location based on your choice.

Configuring multiple IdP servers

Once you have placed all the metadata files, you can expose them to Adeptia Connect/AIS by following the steps given below.

For a clustered set up, you can repeat the steps given in every node of the environment.

1. Open saml.properties file located at **<ConnectPortalInstallFolder>/resources_config/saml**.
2. Uncomment the property **SAML_SSO_IDPS_CONFIGURATION_0_METADATA_LOCATION**.
3. Provide the path of one of the IdP servers metadata file as the value for the property **SAML_SSO_IDPS_CONFIGURATION_0_METADATA_LOCATION** to expose this metadata.
   - If you have placed the metadata file at a location other than ...<ConnectPortalInstallFolder>/resources_config/saml, you need to provide absolute path of the file as the value for the property.
   - The application sets the default IdP based on what metadata file you expose by using the property SAML_SSO_IDPS_CONFIGURATION_0_METADATA_LOCATION. For example, if you provide the path of idp1.xml file as the value for this property, the IdP whose metadata is stored in idp1.xml file becomes the default IdP.

```
#SAML_SSO_SP_CONFIGURATION_REQUIRE_LOGOUT_REQUEST_SIGNED= true
#SAML_SSO_SP_CONFIGURATION_REQUIRE_LOGOUT_RESPONSE_SIGNED= false
#SAML_SSO_METADATA_MANAGER_DEFAULT_IDP=

SAML_SSO_IDPS_CONFIGURATION_0_REGISTRATION_ID = default
#SAML_SSO_IDPS_CONFIGURATION_0_METADATA_LOCATION = file:
${app.working.dir.encoded.path}/resources_config/saml/idp.xml
#SAML_SSO_IDPS_CONFIGURATION_0_REQUEST_TIMEOUT=: 0
#SAML_SSO_IDPS_CONFIGURATION_0_SIGNING_KEY=
```

4. Add the property **SAML_SSO_IDPS_CONFIGURATION_1_METADATA_LOCATION**
5. Provide the path of the another IdP server metadata as the value for the property **SAML_SSO_IDPS_CONFIGURATION_1_METADATA_LOCATION** to expose this metadata.
6. Keep adding the properties by using numbers in incremental fashion in their names, and provide the path of the xml files (metadata files) as their values until all the IdP metadata files are exposed.
   *For example, after you have added **SAML_SSO_IDPS_CONFIGURATION_1_METADATA_LOCATION**, the name of the next property you add should be **SAML_SSO_IDPS_CONFIGURATION_2_METADATA_LOCATION.***

Authenticating a user through non-default IdP server

If you have configured multiple IdPs, the users are by default authenticated through the default IdP. In case you want the user to be authenticated through a non-default IdP, you need to specify the registration Id of that IdP in the application URL as shown below.

Registration Id is the name of provided by the user to the IdP. In case of multiple IdPs, you can use the property SAML_SSO_IDPS_CONFIGURATION_0_REGISTRATION_ID in the saml.properties file to define the registration Id of one of the IdPs, and then replace the number in incremental fashion in the property name to define the registration Id of the next IdP, for example, SAML_SSO_IDPS_CONFIGURATION_1_REGISTRATION_ID, and so on.

```
https://<Domain name or IP>?registrationId=<registration_Id>
```

Landing to a specific page in Adeptia Connect/AIS

In case the users want to land to a specific page in Adetia Connect after getting authenticated through a non-default IdP, they need to enter the application URL in the format as shown in the example below.

```
https://<Domain name or IP>/?idp=<Entity ID of the IdP mentioned in the entityID attribute of its respective idp.xml file>#<dashboard/transactions/allMessages/all>
```

Where,

dashboard/transactions/allMessages/all is the application page where the user may want to land after logging in.

Mapping matching fields of IdP user

User attribute mapping is used for identifying fields in the Service Provider that you want to map with those in the IdP server by synchronizing them on login. It compares the values in the SAML response in case-insensitive manner.

You can map any user field to any arbitrary SAML attribute. For example, you can map the user's username as a Name.

Follow the steps given below to map the fields:

1. Go to the **…<ConnectServerInstallFolder>\AdeptiaServer\ServerKernel\etc\saml** folder.
2. Open **SAMLSSOConfiguration.xml** file in the text editor.
3. Map SAML assertion attributes to Adeptia user fields in the file as shown in the following screenshot.

```
<?xml version="1.0" encoding="UTF-8"?>
<SAMLConfiguration>
    <mapping>
        <field name="email"
            <mapped-attribute>subject.nameid</mapped-attribute>
        </field>
    </mapping>
</SAMLConfiguration>
```

Where,

- **<field>** is the Adeptia user field.
- **<mapped-attribute>** is the SAML assertion attribute.

Once you have mapped fields successfully, next step is to create users in Adeptia Connect/AIS.

User must exist both in SAML and Adeptia Connect/AIS to be authenticated.

Uploading Adeptia Connect/AIS Metadata to IdP Server

Uploading Adeptia Connect/AIS metadata to IdP server allows IdP server to fetch the details (such as server name, metadata information, certificate, encryption, single logout) of Service Provider.

To upload Adeptia Connect/AIS metadata, you need to first download Adeptia Connect/AIS (SP) metadata file, and then upload it to the IdP server. Perform the following steps to upload Adeptia Connect/AIS metadata to IdP Server:

1. Open the browser and hit the URL in the format to download the Adeptia (SP) metadata file.

   URL format for AC v3.7.2:

   `<protocol_name>://<ip_address>:<port_number>/saml2/service-provider-metadata/<registration_Id>`

   URL format for Adeptia Suite:

   `<protocol_name>://<ip_address>:<port_number>/adeptia/saml2/service-provider-metadata/<registration_Id>`

   where,

   <protocol_name> is the name of the protocol, for example, HTTP or HTTPS.

   <ip_address> is the IP address of the computer hosting Adeptia Connect/AIS.

<port_number> is the port number of the computer hosting Adeptia Connect/AIS.

<registration_Id> is the name of the IdP, defined in the saml.properties file, to which you want to import Adeptia metadata.

For example, http://192.168.1.10:8080/saml2/service-provider-metadata/default.

The **spring-<registration_Id>-metadata.xml** file will get downloaded to your computer. This file will have the default metadata information.

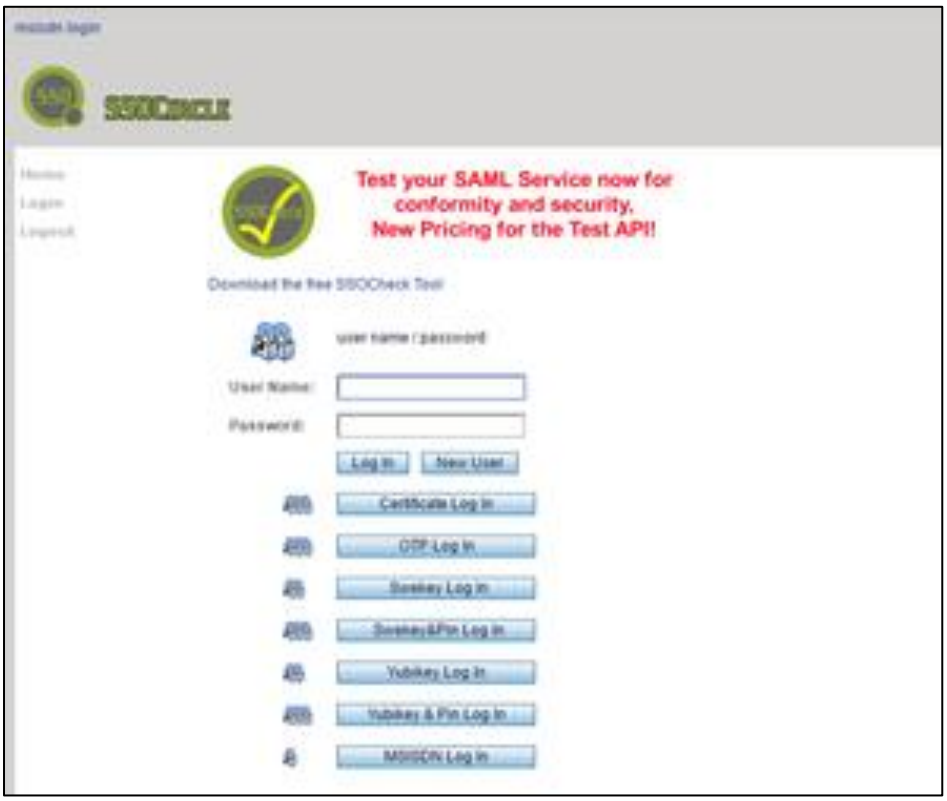
If you want to configure your own SP metadata information in the **spring-<registration_Id>-metadata.xml** file, you need to configure the following properties in the saml.properties file.

Variable Name	Description
SAML_SSO_METADATA_GENERATOR_ENTITY_ID	Unique identifier of the Service Provider. It can be a unique name.
SAML_SSO_METADATA_GENERATOR_ENTITY_BASE_URL	URL to redirect Adeptia Connect/AIS after successful SAML authentication. It needs to be a URL with protocol, server, port, and context path. If you are communicating over SSL protocol, provide the protocol name as https and port number on which Adeptia Connect/AIS is running in the URL.

2. Open the browser, and hit the URL of the IdP.

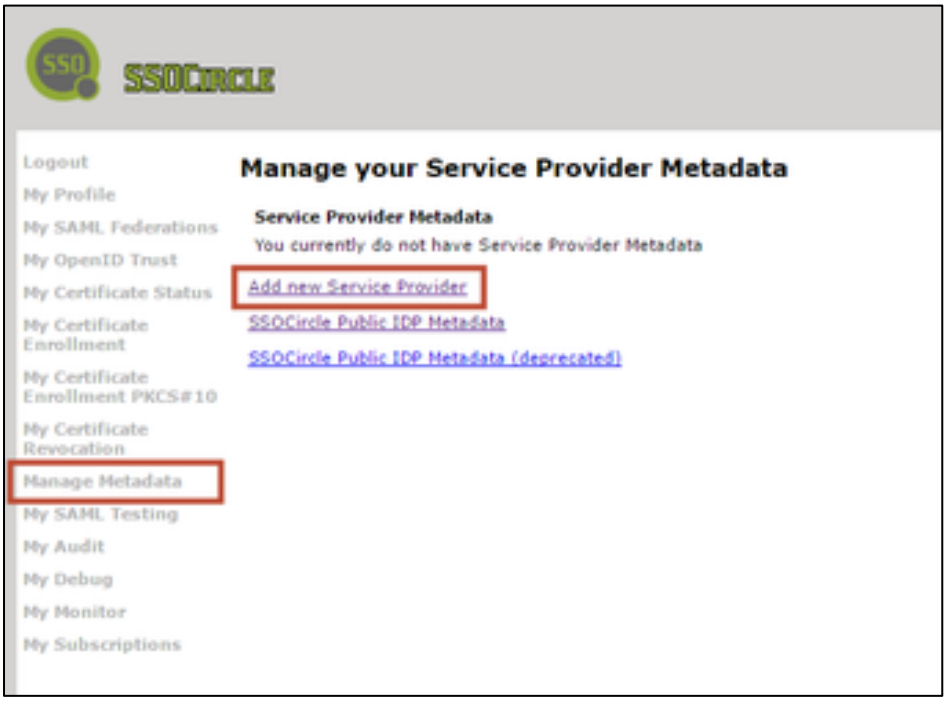
   Depending upon the IdP server that you're using, the login page appears. For example, the screen below depicts the login page of SSOCircle IdP.

   The steps to upload the metadata file may vary from one IdP to another.



3. Type the username and password in the respective fields.
4. Click **Log In**.
5. The **User Profile** screen appears.
6. Click **Manage Metadata** from the left menu options.

7. Click **Add new Service Provider**.



8. Type the name of the Service Provider in the **Enter the FQDN of the ServiceProvider** text box.
9. (Recommended) Select all the attributes (FirstName, LastName, and EmailAddress) in the **Attributes sent in assertion**.
10. Paste the content of Adeptia Connect/AIS (SP) metadata (file downloaded in the first step) in the **Insert your metadata information** text box.
11. Click **Submit** to upload the metadata.

Configuring Adeptia Connect/AIS behind Reverse Proxy/Load Balancer

You can deploy SAML in scenarios where multiple Service Providers process SAML requests forwarded by a reverse-proxy or a load balancer. In order to configure SAML for deployment behind load balancer or reverse-proxy, follow the steps given below:

1. Go to the **...<ConnectPortalInstallFolder>/resources_config/saml**.
2. Open **saml.properties** file in text editor.
3. Set IP address or domain name of the Load Balancer as a value for the property SAML_SSO_METADATA_GENERATOR_ENTITY_BASE_URL in the following format:

   ```
   <protocol_name>://<IP_address or domain_name>
   ```

   Where,

   <protocol_name> is the name of the protocol. For example, http or https.

   <ip_address or domain_name> is the IP address or domain name of the load balancer.

For example, http://www.myserver.com

4. Set the values for the following reverse-proxy/load balancer properties in the saml.properties file.

Property Name	Example value	Description
SAML_SSO_CONTEXT_PROVIDER_LB_SCHEME	http	Name of the scheme (http or https).
SAML_SSO_CONTEXT_PROVIDER_LB_SERVER_NAME	www.myserver.com	Name of the server.
SAML_SSO_CONTEXT_PROVIDER_LB_SERVER_PORT	8080	Port number of the server.
SAML_SSO_CONTEXT_PROVIDER_LB_INCLUDE_SERVER_PORT_IN_REQUEST_URL	false	Whether to include server port number in the URL or not. It must be *false*.
SAML_SSO_CONTEXT_PROVIDER_LB_CONTEXT_PATH	/adeptia	Prefix of a URL path used to select the context(s) to which an incoming request is passed. A URL is in the format: http://hostname.com/**contextPath**/, where each of the path elements can be zero or more separated elements. It must be /adeptia.

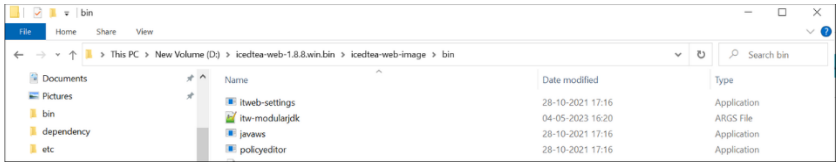
5. Save the file.

Changes for applets to work (for AIS)

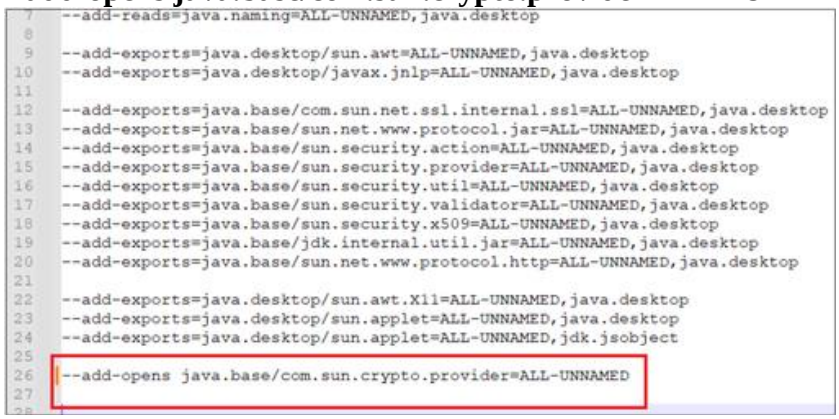
The applets won't open on the client machine. To fix this issue, perform the following actions on the client machine:

1. Upgrade the JRE to v17.
2. Download the **IcedTea-Web** tool zip.
3. Extract the downloaded zip.
4. Go to the path where you have extracted the zip.
   The folder structure would look like the following:

<Drive>:\<Parent Folder (if any)>\icedtea-web-1.8.8.win.bin\icedtea-web-image\bin
(This may vary based on where you have extracted the zip)



5. Double-click **itweb-settings**.
6. On the **IcedTea-Web Control Panel** screen, select **JVM Settings**.
7. Point Java 17 in OpenJDK.
8. Click **Apply**, and then click **OK**.
9. Open itw modulejdk in a Notepad file, paste the following argument, and save the file.
   --add-opens java.base/com.sun.crypto.provider=ALL-UNNAMED



10. Download the data mapper jnlp file that you want open in the applet.
11. Right-click the jnlp file, and then click **Open with** > **Choose another app.**
12. On the **How do you want to open this file?** pop-up screen, do the followings:
    a. Select the **Always use this app to open .jnlp files** checkbox.
    b. Click **More apps.**
    c. Click **Look for another app on this PC.**
13. Browse to the following path.
    …\icedtea-web-1.8.8.win.bin\icedtea-web-image\bin (This path may vary based on where you have extracted the **IcedTea-Web** tool zip).
14. Select the file **javaws.exe** and click **Open**.