

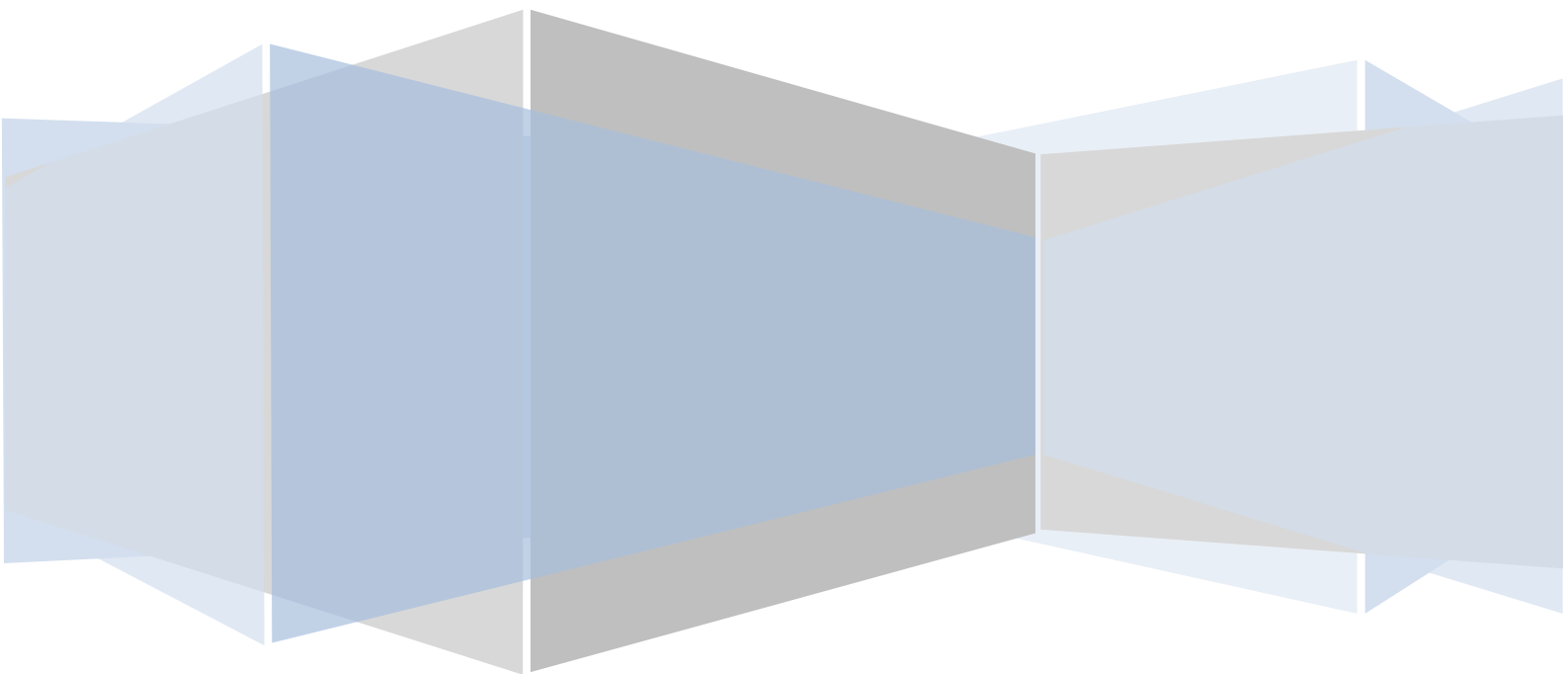
SSO Plugin

Integration for Jasper Server 4

J System Solutions

<http://www.javasystemsolutions.com>

Version 3.5



JSS SSO Plugin - Integration with Jasper Server 4

Introduction.....	3
Jasper Server user administration.....	4
Configuring SSO Plugin.....	5
Installing SSO Plugin for Jasper Server.....	5
Group mapping.....	8
Restricting access.....	8

Introduction

This document covers installation and configuration of SSO Plugin for the popular open source Jasper Server. Separate documents are available for other components (ie BMC AR System, Dashboards).

The JSS support website can be found here:

<http://www.javasystemsolutions.com/jss/support>

There's a video available to assist with installing the SSO Plugin and it can be found at <http://www.javasystemsolutions.com/jss/video>

Jasper Server user administration

Jasper Server maintains its own user database and role mapping. The integration will automatically create users that exist in AR System or Service Manager but do not exist in Jasper Server, and will automatically map ARS/SM administrators to Jasper Server administrators.

Configuring SSO Plugin

You must first set up SSO Plugin to enable the Identity Federation Service:

1. Login to the Mid Tier or Web Tier SSO Plugin setup page.
2. Tick 'Enable Identity Federation Service'.
3. Enter a unique key or press the button to create one. Take a note of the key.
4. Press 'Set configuration' and ensure the SSO Plugin still functions using the 'Test SSO' link.

Installing SSO Plugin for Jasper Server

Please note: JSS support are happy to carry out this procedure for you if preferred.

To enable SSO Plugin for Jasper Server, the following steps must be followed to copy the plugin to Jasper Server:

1. Login to Jasper Server and ensure you know the password for the Administrator account. This will allow you to login manually should you need to troubleshoot the SSO integration once complete.
2. Locate the jasperserver directory in the SSO Plugin installation files. Copy the jss-sso-jasperserver.jar file into the Jasper Server Apache Tomcat WEB-INF/lib directory, typically found in C:\Program Files\Jasper Server\apache-tomcat\webapps\jasperserver\WEB-INF\lib.
3. The Jasper Server web.xml file requires patching. We provide a tool on the JSS support website to do this for you (<http://www.javasystemsolutions.com/jss/service>) and recommend you use it, or you can do this manually by following these steps:
 - 3.1. Open the web.xml.patch file from jasperserver installation directory, select all and copy to clipboard.
 - 3.2. Open the web.xml file from Jasper Server Apache Tomcat directory, ie webapps\jasperserver\WEB-INF\web.xml.
 - 3.3. Paste the block of text copied to clipboard after the XSS filter, ie.

```
<filter>
  <filter-name>XSS</filter-name>
  <filter-
class>com.jaspersoft.jasperserver.war.security.CrossScriptingFilter<
/filter-class>
</filter>
<!-- Patch for SSO Plugin -->
<filter>
  ...
```

3.4. Referring to the patch pasted above, perform the following changes:

- a) **identityFederationServiceURL:** This points to the identity federation service running on the SSO Plugin installation. The identity federation service URL is /jss-sso/identityfederationservice, relative to the Mid Tier or Web Tier installation. Therefore, if the Mid Tier or Web Tier is installed at:

```
http://bmcMid Tier or Web Tier:8080/arsys
```

then the **identityFederationServiceURL** is:

```
http://bmcMid Tier or Web Tier:8080/arsys/jss-sso/identityfederationservice
```

- b) **key:** This must be set to the federated identity key set in the Mid Tier or Web Tier SSO Plugin interface.

4. Open the webapps\jasperserver\WEB-INF\applicationContext-server.xml.

4.1. Locate the following line:

```
<!-- ===== AUTHENTICATION ..
```

and insert the following below it.

```
<bean id="jss.preAuthenticatedAuthenticationProvider"
class="org.springframework.security.providers.preauth.PreAuthenticatedAuthenticationProvider">
  <property name="preAuthenticatedUserDetailsService">
    <bean
class="com.javasystemsolutions.integrations.spring.security.SSOPluginUserDetailsService">
      <property name="sessionFactory" ref="sessionFactory" />
      <property name="userAuthorityService"
ref="jss.txproxy.userAuthorityService" />
    </bean>
  </property>
</bean>
<bean id="jss.j2eefilter"
class="com.javasystemsolutions.integrations.spring.security.SSOPluginPreAuthFilter">
  <property name="authenticationManager"
ref="authenticationManager"/>
</bean>
<bean id="jss.txproxy.userAuthorityService"
class="org.springframework.transaction.interceptor.TransactionProxyFactoryBean">
  <property name="transactionManager" ref="transactionManager"/>
  <property name="target" ref="userAuthorityServiceTarget"/>
  <property name="transactionAttributes">
    <props>
```

```
<prop key="get*">PROPAGATION_REQUIRED</prop>
<prop key="put*">PROPAGATION_REQUIRED</prop>
<prop key="*">PROPAGATION_SUPPORTS</prop>
</props>
</property>
</bean>
```

4.2. Locate the following line:

```
<ref bean="${bean.daoAuthenticationProvider}"/>
```

and insert the following below it.

```
<ref bean="jss.preAuthenticatedAuthenticationProvider"/>
```

5. Open the webapps\jasperserver\WEB-INF\applicationContext-server-web.xml, and locate the following string:

```
/**=httpSessionContextIntegrationFilter,
```

and append the following (in bold):

```
/**=httpSessionContextIntegrationFilter, jss.j2eefilter,
```

6. Restart the Jasper Server Tomcat instance.

7. Navigate to: <http://host/jasperserver> You should now be logged in as the AR System or Service Manager user to which your SSO user is mapped.

Group mapping

By default, the following AR System and Service Manager groups are mapped to a Jasper Server administrator user: Administrator, Incident Config, Asset Config, Change Config, Problem Config, Release Config, SLM Config, SRM Administrator.

You can specify a custom group mapping from ARS/SM ITSM to Jasper Server by editing the SSOPluginUserDetailsService bean configured in the Spring applicationContext-security.xml file. This example shows how Problem Manager can be mapped to the Jasper Server role ROLE_PROBLEM_MANAGER (note, the ROLE_ is not specified in the mapping, but ROLE_PROBLEM_MANAGER must be setup in Jasper Server):

```
<bean
class="com.javasystemsolutions.integrations.spring.security.SSOPluginUserDetailsService">
  <property name="sessionFactory" ref="sessionFactory" />
  <property name="userAuthorityService"
ref="jss.txproxy.userAuthorityService" />
  <property name="groupMapping"><map>
    <entry key="Problem Manager">
      <value>PROBLEM_MANAGER</value>
    </entry>
  </map></property>
</bean>
```

The administrator groups can also be configured by adding the following to the bean:

```
<property name="adminGroups"><list>
  <value>Group name</value><value>Another group name</value>
</list></property>
```

Jasper Server must be restarted when this file is changed.

Restricting access

If you want to restrict SSO access to Jasper Reports, a set of groups (of which a user must be a member of at least one) can be configured. To do this, locate the SSOPluginPreAuthFilter and add the text in bold:

```
<bean id="jss.j2eefilter"
class="com.javasystemsolutions.integrations.spring.security.SSOPluginPreAuthFilter">
  <property name="authenticationManager"
ref="authenticationManager"/>
  <property name="allowGroups"><b><list>
```

```
<value>Reporting Group</value>  
</list></property>  
</bean>
```