

Appendix A: Configuring Custom Jasper Roles

This appendix contains information that allows you to make a custom Jasper role work as per the defined logic.

Jasper Roles Overview

The JasperReports Server comes with the following roles: Role_superuser, Role_administrator, Role_poweruser, Role_author and Role_user. These roles can be assigned to your users to provide access to various available features in the JasperReports Server. You can assign these pre-configured Jasper roles to RiskVision roles. You may also create custom Jasper roles in the JasperReports Server and assign them to RiskVision roles.

Note: This technical document assumes that you are familiar with the information about creating a custom role in the JasperReports Server and assigning a Jasper role to an role in the RiskVision application. For more information, please refer to the section, Assigning JasperReports Server Role.

File Settings for a Custom Jasper Role

These are the files in which the settings are required:

- `actionmodel-navigation.xml`
- `applicationContext-rest-services.xml`

The following scenario illustrates how to modify the file settings.

Scenario: A custom Jasper role allows a user to create an ad hoc view and a report in the JasperReports Server. Here, the Jasper role is referred as 'CustomJasperRole.'

To make a custom Jasper role function in accordance with your requirements, please follow the steps given below:

1. On the JasperReports Server host, go to the `%JASPERREPORTS_HOME\ReportServer\apache-tomcat\webapps\jasperserver-pro\WEB-INF`, open the `actionmodel-navigation.xml` file using a text editor.
2. Locate the context element and add the custom Jasper role to the appropriate lines of code depending on the custom Jasper role functionality. For this scenario, add the custom Jasper role to the `testArgs` attribute within the context element. And then, add the custom Jasper role to the `testArgs` attribute within the element that is beneath the other element in which the value of `testArgs` attribute is **AHD**. The following graphic demonstrates the code changes.

```
<!--jasperserver:designer:pro:only-->
<context name="/main_create_action" test="checkAuthenticationRoles"
testArgs="ROLE_ADMINISTRATOR,ROLE_POWERUSER|Agilience,ROLE_AUTHOR|Agilience,CustomJasperRole|Agilience">
  <condition test="!isAdminRole">
    <selectAction labelKey="NAV_005_CREATE">
      <condition test="!isAvailableProFeature" testArgs="AHD">
        <condition test="checkAuthenticationRoles"
testArgs="ROLE_ADMINISTRATOR,ROLE_POWERUSER|Agilience,ROLE_AUTHOR|Agilience,CustomJasperRole|Agilience">
          <option labelKey="NAV_051_ADHOC_REPORT" action="primaryNavModule.navigaticionOption"
actionArgs="designer"/>
          <option labelKey="NAV_057_REPORT" action="JRS.CreateReport.selectADV"
actionArgs="createReport"/>
        </condition>
      </condition>
    </selectAction>
  </condition>
</context>
```

3. Locate the bean element within the bean element and add the custom Jasper role to the appropriate lines of code depending on the custom Jasper role functionality. For this scenario, add the following code beneath the element within the element of other element in which the value attribute is **SEE_ADHOCS_ALLOWED**.

```
<value>
```

CustomJasperRole|

```
</value>
```

The following graphic demonstrates the code changes:

```
<bean class='com.jaspersoft.jasperserver.jaxrs.poc.hypermedia.workflow.service.WorkflowsSecurityFilter'>
  <property name="supportedAttribute" value="SEE_ADHOCS_ALLOWED">
  <property name="allowedRoles">
    <util:list>
      <value>ROLE_ADMINISTRATOR</value>
      <value>ROLE_POWERUSER|Agilience</value>
      <value>ROLE_ADHOC|Agilience</value>
      <value>CustomJasperRole|Agilience</value>
    </util:list>
  </property>
  <property name="workflowsToSecure">
    <list>
      <value>adhocView</value>
    </list>
  </property>
</bean>
```

4. Restart the Jasper Tomcat service.