How to Use Preconditions in Rules Composer

Applies to:
SAP NetWeaver Business Rules Management. For more information, visit the Business Rules Management homepage.

Summary
This is a short tutorial on how to work with Preconditions in the Rules Composer. A business use case has been created for the purpose.

Given details such as connection provider, type of line/connection and connection destination, the long distance phone rate must be calculated.

This tutorial guides you to create rules and add precondition to a rule to calculate the long distance phone rate given a set of criteria.

Author: Arti Gopalan and Rupa Rajagopalan
Company: SAP Labs India
Created on: 26 May 2008
Table of Contents

Prerequisites ....................................................................................................................................................... 3

Knowledge Required ........................................................................................................................................ 3

Software Requirements ................................................................................................................................... 3

Procedure ........................................................................................................................................................... 3

Creating the Rules Composer DC .................................................................................................................. 3

Creating the Ruleset ....................................................................................................................................... 3

Creating the XML Schema .............................................................................................................................. 4

Adding XSD Elements ................................................................................................................................................. 5

Creating the Rules .......................................................................................................................................... 6

Creating the SetLine1CallCharge rule ......................................................................................................................... 6

Creating the ChkIfIntraCountry rule ........................................................................................................................... 10

Adding the Precondition ................................................................................................................................ 11

Deploying the Rules ...................................................................................................................................... 12

Executing the Rules ...................................................................................................................................... 13

Creating the Web Module ............................................................................................................................... ........ 13

Adding Dependency to the Web Module .................................................................................................................... 13

Creating the Enterprise Application ........................................................................................................................ 14

Adding Dependency to the Enterprise Application .................................................................................................. 14

Creating the application.xml ............................................................................................................................... ........ 14

Building and Deploying ................................................................................................................................ 15

Running the Web Module ............................................................................................................................... ............ 15

Related Content ................................................................................................................................................ 18

Copyright ........................................................................................................................................................... 19
Prerequisites

Knowledge Required

• You have basic knowledge in rules modeling
• You are familiar with Business Rules Management System

Software Requirements

• You work in the SAP NetWeaver Developer Studio
• Your SAP NetWeaver Developer Studio version includes the Rules Composer perspective
• You should have a running instance of SAP AS, and should have configured the SAP NetWeaver Developer Studio with this instance

Note: In the SAP NetWeaver Developer Studio, choose Window -> Open Perspective -> Other. In the dialog box that appears, choose Rules Composer and choose OK.

Procedure

Creating the Rules Composer DC

1. In the SAP NetWeaver Developer Studio, choose File -> New -> Project.
2. In the wizard that appears, expand the Rules Composer node and choose Rules Composer Development Component. Choose Next,
3. In the screen that appears, choose the software component where you want to create the DC. For example the software component could be MyComponents [demo.sap.com] under the Local Development node. Choose Next.
4. In the screen that appears, enter callchargecalc in the Name field and choose Finish.

You should see the Rules Composer DC: callchargecalc node in the Project Explorer view.

Creating the Rule Set

1. In the Project Explorer view, expand the Rules Composer DC: callchargecalc node and in the context menu of the Rules Modeling node, choose New Ruleset.
2. In the dialog box that appears, enter CallChargeCalc in the field. Choose OK.
You should see the ruleset: CallChargeCalc under the Rules Modeling node as shown below:

Creating the XML Schema

1. In the Project Explorer view, expand the src node and in the context menu of the wsdl node, choose New -> Other.
2. In the wizard that appears, expand the XML node and choose XML Schema. Choose Next.
3. In the screen that appears, enter callcharges.xsd in the File Name field. Choose Finish.
4. In the Project Explorer view, you should see the callcharges.xsd under the wsdl node.
5. Double-click the callcharges.xsd and in the window that appears choose the Source tab at the bottom.

In the tab page that appears delete all existing content and copy the following in the Source tab page:

```xml
<?xml version="1.0" encoding ="UTF-8"?>
<xsd:schema xmlns:xsd ="http://www.w3.org/2001/XMLSchema"
xmlns="http://www.example.org/CallCharges" elementFormDefault="qualified"
targetNamespace="http://www.example.org/CallCharges">
  <xsd:element name="CallCharges" type="CallChargesType"/>
  <xsd:complexType name="CallChargesType">
    <xsd:sequence maxOccurs="unbounded">
      <xsd:element name="Charges" type="CallChargeType"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="CallChargeType">
    <xsd:sequence>
      <xsd:element name="ConnectionProvider" type="xsd:string"/>
      <xsd:element name="ConnectionRate" type="xsd:double"/>
      <xsd:element name="DestinationCountry" type="xsd:string"/>
      <xsd:element name="OriginCountry" type="xsd:string"/>
      <xsd:element name="TypeofLine" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:schema>
```
6. Press Ctrl+Shift+F.
7. Save the changes.

Adding XSD Elements
1. In the Project Explorer view, expand the Rules Modeling node and double-click the Aliases node.
2. In the Project Aliases Editor that appears, choose the XSD Aliases tab and in the tab page that appears, choose the Add XSD Elements tab.
3. In the dialog box that appears, expand the http://www.example.org/CallCharges node and choose Call Charges.
5. In the Alias Name table select all the XML schema element checkboxes.

In the Alias Name table, click each of the aliases. The aliases become editable. Enter an alternative name for the alias.

<table>
<thead>
<tr>
<th>Alias Name</th>
<th>Rename as</th>
</tr>
</thead>
<tbody>
<tr>
<td>CallCharges/Charges/ConnectionProvider</td>
<td>Connection Provider</td>
</tr>
<tr>
<td>CallCharges/Charges/ConnectionRate = {double}</td>
<td>Connection Rate = {double}</td>
</tr>
<tr>
<td>CallCharges/Charges/DestinationCountry</td>
<td>Destination Country</td>
</tr>
<tr>
<td>CallCharges/Charges/TypeofLine</td>
<td>TypeofLine</td>
</tr>
</tbody>
</table>

The result must be as shown below:
Creating the Rules

Creating the SetLine1CallCharge rule

1. In the Project Explorer view, expand the Rules Composer DC:callchargecalc node, the Rules Modeling node and in the context menu of the ruleset:CallChargeCalc node, choose New Rule.

2. In the dialog box that appears, enter SetLine1CallCharge in the field and choose OK.

In the Outline view, the SetLine1CallCharge node appears as shown below:

3. In the Rule Editor that appears, under If section, choose the Add icon.

4. The default condition: Operation.isSuccessful Equals true appears.

5. Edit the default condition as follows:
a. Choose the LValue: `Operation.isSuccessful` and in the drop down menu choose `TypeOfLine` as shown below:

b. Leave the comparator: `Equals` as it is.
c. Choose the RValue: Default Value and in the inline textbox enter Line1.
   The result must be as shown below:

   ![Image of Rules Composer interface]

6. Under Then section, choose the Add icon and in the drop down menu that appears, choose Execute <Method Name>.
7. Choose <Action Method> and in the drop down menu choose Connection Rate.

8. Choose (double) and in the inline textbox enter 100 as shown below:

9. Save the changes.

The result must be as shown below:
Creating the ChkIfIntraCountry rule

1. In the **Project Explorer** view, expand the Rules Composer DC:callchargecalc node, the **Rules Modeling** node and in the context menu of the ruleset: CallChargeCalc node, choose **New Rule**.

2. In the dialog box that appears, enter ChkIfIntraCountry in the field and choose **OK**. In the **Outline** view, the ChkIfIntraCountry node appears.

3. In the Rule Editor that appears, under **If** section, choose the **Add** icon.

4. The default condition: *Operation.isSuccessful Equals true* appears.

5. Edit the default condition as follows:
   a. Choose the LValue: *Operation.isSuccessful* and in the drop down menu choose *Destination Country*.
   b. Leave the comparator: *Equals* as it is.
   c. Choose the RValue: *Default Value* and in the drop down choose *Origin Country*.

6. Under **Then** section, choose the **Add** icon and in the drop down menu that appears, choose **Execute-<Method Name>**.

7. Choose **<Method Name>** and in the drop down menu choose *Connection Rate*.

8. Choose *{double}* and in the inline textbox enter 50.

9. Save the changes.

The result must be as shown below:
Adding the Precondition

1. In the Outline view, choose the SetLine1CallsCharge node.
2. Under Preconditions: section, choose the Add icon and in the drop down menu that appears, choose ChkIfIntraCountry as shown below:
3. By default the rule is set to must be satisfied. Change it to must not be satisfied by clicking on it as shown below:

Deploying the Rules

1. In the Project Explorer view, in the context menu of the Rules Composer DC:callchargecalc node, choose Development Component -> Build.

2. In the dialog box that appears, make sure the callchargecalc checkbox is selected and choose OK.

   Choose Window -> Show View -> Other and in the dialog box that appears, expand the Development Infrastructure node and double-click Infrastructure Console to check if the build has happened successfully.

3. In the context menu of the Rules Composer DC:callchargecalc node, choose Development Component -> Deploy.

4. In the dialog box that appears, make sure the callchargecalc checkbox is selected and choose OK.

5. Open the Infrastructure Console, to check if the deploy has happened successfully.
How to Use Preconditions in Rules Composer

Executing the Rules

Creating the Web Module

1. In the SAP NetWeaver Developer Studio, choose File -> New -> Project.
2. In the wizard that appears, expand the Development Infrastructure node and choose Development Component. Choose Next.
3. In the screen that appears, expand the J2EE node and choose Web Module. Choose Next.
4. In the screen that appears, choose the software component where you want to create the DCs. For example expand the Local Development node and choose MyComponents [demo.sap.com]. Choose Next.
5. In the screen that appears, enter callchargecalculator_wm in the Name field. Choose Finish.
6. The Java EE perspective opens and in the Project Explorer view, you should see the callchargecalculator_wm node.

Note: If the Java EE perspective does not open, in the SAP NetWeaver Developer Studio choose Window -> Open Perspective -> Other. In the dialog box that appears, choose Java EE and choose OK.

Adding Dependency to the Web Module

1. Choose Window -> Open Perspective -> Other.
2. In the dialog box that appears, choose Development Infrastructure. Choose OK.
3. If the Component Browser view is not open, choose Window -> Show View -> Other.
4. In the dialog box that appears, expand the Development Infrastructure node and choose Component Browser. Choose OK.
5. In the Component Browser view, expand the MyComponents[demo.sap.com] node and choose the callchargecalculator_wm node.

Note: If the Component Properties view does not open, choose Window -> Show View -> Other. In the dialog box that appears, expand the Development Infrastructure node and choose Component Properties. Choose OK.

6. In the Component Properties view, choose Dependencies.
7. Choose the Add button and in the wizard that appears, expand the BRMS-FACADE[sap.com] node and select the tc/brms/facade checkbox. Choose Next.
8. In the screen that appears, select the Design Time, Deploy Time, Run Time checkboxes. Choose Finish.

Note: Make sure you are in the Java EE perspective.

Unzip the project file and do the following:

1. Expand the web module: callchargecalculator_wm node and in the context menu of the source node, choose New -> Other.
2. In the wizard that appears, expand the Java node and choose Package. Choose Next.
3. In the screen that appears, enter com.sap.helper in the Name field.
6. Expand the Web Content node under the callchargecalculator_wm node and copy the following files into it: CallCharges.jsp, index.jsp and invoker.jsp
Creating the Enterprise Application

1. In the SAP NetWeaver Developer Studio, choose File -> New -> Project.
2. In the wizard that appears, expand the Development Infrastructure node and choose Development Component. Choose Next.
3. In the screen that appears, expand the J2EE node and choose Enterprise Application. Choose Next.
4. In the screen that appears, choose the software component where you want to create the DCs. For example expand the Local Development node and choose MyComponents [demo.sap.com]. Choose Next.
5. In the screen that appears, enter callchargecalc_ear in the Name field. Choose Next.
6. Skip the screen that appears.
7. In the New EAR Project screen select the LocalDevelopment~LocalDevelopment~callchargecalculator_wm~demo.sap.com checkbox.

In the Project Explorer view, you should see the callchargecalc_ear node.

Adding Dependency to the Enterprise Application

1. In the Component Browser view, expand the MyComponents[demo.sap.com] node and choose the callchargecalc_ear node.
2. In the Component Properties view, choose Dependencies.
3. Choose the Add button and in the wizard that appears, expand the BRMS-FACADE[sap.com] node and select the tc/brms/facade checkbox. Choose Next.
4. In the screen that appears, select the Design Time, Deploy Time, Run Time checkboxes. Choose Finish.
5. Choose the Add button again and in the wizard that appears, expand the My Components[demo.sap.com] node and select the callchargecalculator_wm checkbox. Choose Next.

Note: In the context menu of the callchargecalculator_wm and callchargecalc_ear nodes, choose Sync/Create Project Sync Used DCs. In the dialog box that appears, choose OK.

Creating the application.xml

Make sure that you are in the Java EE perspective.

1. Expand the enterprise application: callchargecalc_ear node and in the context menu of the Deployment Descriptor: LocalDevelopment~LocalDevelopment~callchargecalc_ear~demo.sap.com node, choose create application.xml.

You should see the application.xml window with the following lines:

```xml
<?xml version="1.0" encoding="ASCII"?>
  <display-name>LocalDevelopment~LocalDevelopment~callchargecalc_ear~demo.sap.com</display-name>
  <module>
    <web>
      <web-uri>demo.sap.com~callchargecalculator_wm.war</web-uri>
    </web>
  </module>
</application>
```
How to Use Preconditions in Rules Composer

2. Replace `<context-root>LocalDevelopment~LocalDevelopment~callchargecalculator_wm~demo.sap.com</context-root>` with `<context-root>Call Charges Calculator</context-root>`

Note: Instead of `LocalDevelopment~LocalDevelopment~callchargecalculator_wm~demo.sap.com`, you need to enter the customized application name that is in this tutorial, the name of the application is Call Charges Calculator.

Building and Deploying

Make sure that you are in the Java EE perspective.

1. In the context menu of the `callchargecalculator_wm` and `callchargecalc_ear` nodes, choose Build.
2. In the dialog box that appears, choose OK.
3. In the context menu of the `callchargecalculator_wm` and `callchargecalc_ear` nodes, choose Deploy.
4. In the dialog box that appears, choose OK.
5. Open the Infrastructure Console, to check if the build and deploy actions have happened successfully.

Running the Web Module

1. Open the browser and enter the Application Server address followed by port name and the Web Module Name.
2. Enter the following data in the respective fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>User Entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection Provider</td>
<td>A</td>
</tr>
<tr>
<td>Destination Country</td>
<td>India</td>
</tr>
<tr>
<td>Origin Country</td>
<td>India</td>
</tr>
<tr>
<td>Type of Line</td>
<td>Line 1</td>
</tr>
</tbody>
</table>

3. Choose Submit.

You should get the Call Charges as 50 because IntraCountry rule gets satisfied.
Here is a snapshot of the web module:

![Call charge calculator](http://10.66.212.190:50000/docs/CallChargeCalculate.jsp)

**BRMS Invocation Client**

- **Connection Provider (Providers - A, B):** A
- **Destination Country (Countries - India, USA, Germany):** India
- **Origin Country (Countries - India, USA, Germany):** India
- **Type of Line (Lines - Line 1, Line 2):** Line 1

**Call Charges:**

50.0

**Also try this:**

1. Enter the following data in the respective fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>User Entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection Provider</td>
<td>A</td>
</tr>
<tr>
<td>Destination Country</td>
<td>India</td>
</tr>
<tr>
<td>Origin Country</td>
<td>USA</td>
</tr>
<tr>
<td>Type of Line</td>
<td>Line 1</td>
</tr>
</tbody>
</table>

2. Choose **Submit**.

You should get the Call Charges as 100 because **IntraCountry** rule does not get satisfied and Type of Line is Line 1.
Here is a snapshot of the Web Module:

If you enter Line 2 in the Type of Line field, the Call Charges will be set to 0 because no rule has been written to capture this case.

Here is the snapshot of the Web Module:
Related Content

For more information, visit the Business Rules Management homepage.