How to Use Decision Tables 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 Decision Tables 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 a decision table to calculate the long distance phone rate given a set of criteria.

Authors: Arti Gopalan and Rekha Narasapuram
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
Renaming XSD Aliases ........................................................................................................................................5
Creating the Decision Table ............................................................................................................................6
Adding Condition and Action Values ..............................................................................................................7
Creating the Rule ............................................................................................................................................8
Deploying the Rules ......................................................................................................................................11
Executing the Rules ......................................................................................................................................11
Creating the Web Module .................................................................................................................................11
Adding Dependency to the Web Module ..........................................................................................................12
Creating the Enterprise Application ...............................................................................................................12
Adding Dependency to the Enterprise Application ..........................................................................................13
Creating the application.xml ...........................................................................................................................13
Building and Deploying .................................................................................................................................14
Running the Web Module ...............................................................................................................................14

Related Content ................................................................................................................................................17

Copyright...........................................................................................................................................................18
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 callchargecalc node in the Project Explorer view.

Creating the Ruleset

1. In the Project Explorer view, expand the callchargecalc node and in the context menu of the Rules Modeling node, choose New Ruleset.
2. In the dialog box that appears, enter SetCallCharge in the field. Choose OK.
You should see the ruleset: `SetCallCharge` 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. In the `callcharges.xsd` 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'?>
   <xs:schema xmlns:xs='http://www.w3.org/2001/XMLSchema'
   xmlns='http://www.example.org/CallCharges' elementFormDefault='qualified'
   targetNamespace='http://www.example.org/CallCharges'>
   <xs:element name='CallCharges' type='CallChargesType'/>
   <xs:complexType name='CallChargesType'>
   <xs:sequence maxOccurs='unbounded'>
   <xs:element name='Charges' type='CallChargeType'/>
   </xs:sequence>
   </xs:complexType>
   </xs: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 select Call Charges.
5. Under the Alias Name section, select all the XML schema element checkboxes.

Renaming XSD Aliases
Under the Alias Name section, 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 follows:
Creating the Decision Table
You can capture the criteria and the corresponding rates for long distance phone calls in a single table.

1. In the Project Explorer view, expand the callchargecalc node, the Rules Modeling node and in the context menu of the SetCallCharge node, choose New Decision Table.
2. In the Decision Table Creation Wizard that appears, enter CallChargeDT in the Decision Table Name field and choose Next.
3. On the Select the Conditions screen, double-click the following aliases in the Available Conditions section:
   - Destination Country
   - Origin Country
The aliases appear in the Selected Conditions section as follows:

4. Choose Next.
5. On the Select the Actions screen, double-click ConnectionRate = {double} in the Available Actions section.
How to Use Decision Tables in Rules Composer

The alias appears in the **Selected Actions** section as follows:

6. Choose **Finish**.

**Adding Condition and Action Values**

1. In the Decision Table Editor, double-click each of the cells under the condition headers (**Origin Country** and **Destination Country**) and enter the values.

**Note:** You can also choose **Add Condition Value** in the context menu of the cells under the condition headers and in the dialog box that appears enter all the values in the available rows.

2. Double-click each cell under the action header (**ConnectionRate = {double}**) in the Decision Table, enter a value and press **Enter key**.

3. Save the changes.
Refer to the Decision Table below for the condition and action values you need to enter:

![Decision Table Image]

**Creating the Rule**

You need to create a rule called *CallChargeRule* that evaluates the *CallChargeDT*.

1. In the *Project Explorer* view, expand the *callchargecalc* node, the *Rules Modeling* node and in the context menu of the *SetCallCharge* node, choose *New Rule*.
2. In the dialog box that appears, enter *CallChargeRule* in the field and choose *OK*.
3. In the Rule Editor that appears, under *If* section, choose the *Add* icon.
   The default condition: *Operation.isSuccessful Equals true* appears.
4. Edit the default condition as follows:
   a. Choose the LValue: \textit{Operation.isSuccessful} and in the drop down menu choose \textit{Connection Provider} as shown below:
   b. Leave the comparator: \textit{Equals} as it is
   c. Choose the RValue: \textit{Default Value} and in the inline textbox enter \textit{A}. 
The result must be as shown below:

5. Under *Then* section, choose the *Add* icon and in the drop down menu that appears, expand the *Evaluate-Decision Table* node and choose *CallChargeDT* as shown below:
The result must be as shown below:

6. Save the changes.

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, select the callchargecalc checkbox and choose OK.
   
   To check if the build has happened successfully, check the Infrastructure Console.

   **Note:** If the Infrastructure Console is not open, choose Window -> Show View -> Other and in the dialog box that appears, expand the Development Infrastructure node and choose Infrastructure Console and then choose OK.

3. In the context menu of the Rules Composer DC:callchargecalc node, choose Development Component -> Deploy.
4. In the dialog box that appears, select the callchargecalc checkbox and choose OK.

   **Note:** Open the Infrastructure Console, to check if the deploy has happened successfully.

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.

**Adding Dependency to the Web Module**

Make sure that you are in the **Development Infrastructure** perspective.

1. In the **Component Browser** view, expand the **Local Development** node, **MyComponents[demo.sap.com]** node and choose the **callchargecalculator_wm** node.

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

2. In the **Component Properties** view, choose the **Dependencies** tab.

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**.

**Note:** In the context menu of the **callchargecalculator_wm** node, choose **Sync / Create Project > Sync Used DCs**. In the dialog box that appears, choose **OK**.

Unzip the project file and do the following

Make sure that you are in the **Java EE** perspective.

1. Expand the web module: **callchargecalculator_wm node** and in the context menu of the **Java Resources: 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.
4. Choose **Finish**.
5. Copy **EngineInvoker.java** file into **com.sap.helper**.
6. In the context menu of the **Web Content** node, copy the following files: **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. Choose **Next**.
7. In the screen that appears, select the **LocalDevelopment~LocalDevelopment~callchargecalculator_wm~demo.sap.com** checkbox. Choose **Finish**.
8. In the **Project Explorer** view, you should see the **callchargecalc_ear** node.
Adding Dependency to the Enterprise Application

Make sure that you are in the Development Infrastructure perspective.

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 the Dependencies tab.
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.

Note: In the context menu of the callchargecalc_ear node, 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~callchargecalc_ear~demo.sap.com node, choose create application.xml

Note: Expand the enterprise application: callchargecalc_ear node, META-INF node double click the application.xml node.

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

```xml
<?xml version = "1.0" encoding = "ASCII"?>
<application xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns="http://java.sun.com/xml/ns/javaee"
    xmlns:application="http://java.sun.com/xml/ns/javaee/application_5.xsd"
    <display-name>LocalDevelopment~LocalDevelopment~callchargecalc_ear~demo.sap.com</display-name>
    <module>
        <web>
            <web-uri>demo.sap.com~callchargecalculator_wm.war</web-uri>
            <context-root>LocalDevelopment~callchargecalculator_wm~demo.sap.com</context-root>
        </web>
    </module>
</application>
```

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

3. Save the changes.

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 CallChargeCalc.
How to Use Decision Tables in Rules Composer

Building and Deploying

Make sure that you are in the Development Infrastructure perspective.

1. In the Component Browser view, expand the MyComponents[demo.sap.com] and 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 callchargecalc_ear node, 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.

Note: You can also build and deploy in the Java EE perspective.

1. In the Project Explorer view, in the context menu of the callchargecalculator_wm and callchargecalc_ear nodes, choose Development Component > Build.

2. In the dialog box that appears, choose OK.

3. In the context menu of the callchargecalc_ear node, choose Development Component > Build.

4. In the dialog box that appears, choose OK.

Running the Web Module

1. Open the browser and enter the Application Server Address followed by the port number and the application name: CallChargeCalc.

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>USA</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 200 because in the CallChargeDT decision table, the call charge from India to USA has been set to 200.
Here is the snapshot of the web module:

![BRMS Invocation Client screenshot]

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>Germany</td>
</tr>
<tr>
<td>Origin Country</td>
<td>India</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 150 because in the CallChargeDT decision table, the call charge from India to Germany has been set to 150.
Here is the snapshot of the web module:

![BRMS Invocation Client](image)

**Note:** Even if you enter the Type of Line as Line 2, the Call Charges will remain the same.
Related Content
For more information, visit the Business Rules Management homepage.