



**WebSphere®** Monitor

Version 6.0.2



## **Generating a Monitor Model using the BMV and BPEL model**

## Table of Contents

|   |    |
|---|----|
| Part 1. Overview .....  | 5  |
| Sample project interchange file and enterprise archive files .....    | 5  |
| Software requirements.....  | 6  |
| KPIs required for the Clip and Tacks application .....                | 6  |
| Part 2. Generating monitor models from BMV and the BPEL process ..... | 12 |
| Exporting the monitor model from BMV .....                            | 12 |
| Generating the monitor model from the BPEL process .....              | 14 |
| Part 3. Using the dual-model approach .....                           | 21 |
| Defining additional data needed by the KPI model .....                | 22 |
| Defining additional events to be sent to the KPI model .....          | 33 |
| Setting up the Data Mart model .....                                  | 58 |
| Setting up the KPI model .....  | 60 |
| Setting up the visual model for the KPI Model.....                    | 67 |
| Part 4. Using the merged-model approach .....                         | 68 |
| Defining additional data needed for the KPI.....                      | 68 |
| Setting up the Data Mart model .....                                  | 78 |
| Setting up the KPI model .....  | 80 |
| Part 5. Deciding which approach to use .....                          | 87 |
| Part 6. How to run the sample .....                                   | 88 |

## Table of Figures

|  |    |
|--|----|
| Figure 1. KPIs for the Average Order Fulfillment Time metric – figure 1..... | 9  |
| Figure 2. KPIs for the Average Order Fulfillment Time metric – figure 2..... | 10 |
| Figure 3. Clips and Tacks sample, as shown in the BMV tool .....             | 11 |
| Figure 4. Select WebSphere Business Monitor Development Toolkit (.mm).....   | 12 |
| Figure 5. Select Export specific elements and the process to export .....    | 13 |
| Figure 6. Click File ► Import ► Project Interchange .....                    | 14 |
| Figure 7. Select projects with the exception of “ClipsAndTacksSample” .....  | 15 |
| Figure 8. Select the OrderHandlingFuture1 process .....                      | 16 |
| Figure 9. Create a new monitor project or select an existing one .....       | 17 |
| Figure 10. Process selection Entry and Exit events .....                     | 18 |
| Figure 11. Entry and Exit events for the activities .....                    | 19 |

|   |    |
|---|----|
| Figure 12. BPELVariable entries.....  | 20 |
| Figure 13. KPI model (ClipsAndTackKPIIMM) generated from BMV .....                              | 21 |
| Figure 14. BPEL-generated model (ClipsAndTacksSample).....                                      | 22 |
| Figure 15. BPEL model with the metrics, counter, and stopwatch .....                            | 23 |
| Figure 16. Example of the new trigger.....  | 24 |
| Figure 17. Map for the City metric displayed with vertical orientation.....                     | 25 |
| Figure 18. Metric Value Maps values displayed for New Order Trigger.....                        | 28 |
| Figure 19. Define the map for the Order Date metric .....                                       | 29 |
| Figure 20. Metric with the added metric maps for the triggers.....                              | 30 |
| Figure 21. Ship Percentage metric with its map.....   | 31 |
| Figure 22. Declined Order Counter trigger counter .....   | 32 |
| Figure 23. OrderHandlingFuture1 Termination Trigger settings .....                              | 33 |
| Figure 24. Definition of the NewOrderEvent.....   | 34 |
| Figure 25. Definition of the ApprovalStepUpdate.....  | 35 |
| Figure 26. Definition of the CompleteOrder.....   | 35 |
| Figure 27. Definition of the DeclinedOrderCtrUpdate .....                                       | 35 |
| Figure 28. Definition of the ReceiveOrder.....  | 36 |
| Figure 29. Definition of ShipPercentageUpdate.....  | 36 |
| Figure 30. Definition of StatusUpdate.....  | 36 |
| Figure 31. The StatusUpdate trigger .....   | 37 |
| Figure 32. The CountryUpdate_Trigger .....  | 38 |
| Figure 33. The OrderAmountUpdate_Trigger.....   | 39 |
| Figure 34. The New_Order_To_KPI outbound event definition .....                                 | 40 |
| Figure 35. The Order_Completed_To_KPI outbound event definition .....                           | 41 |
| Figure 36. The Receive_Order_To_KPI outbound event definition.....                              | 42 |
| Figure 37. Ship_Percentage_Update_To_KPI values set by defaults or inbound events .....         | 43 |
| Figure 38. Update_Approval_Step_Completed values set by defaults or inbound events.....         | 44 |
| Figure 39. Update_Status event values set by default values or inbound events .....             | 45 |
| Figure 40. Triggers that cause the Declined_Order_Ctr_Update event.....                         | 46 |
| Figure 41. Triggers that cause the Update_Country event.....                                    | 47 |
| Figure 42. Triggers that cause the Update_OredrAmount event .....                               | 48 |
| Figure 43. Change the ID and Name to make them easier to read.....                              | 49 |
| Figure 44. Model with the default metrics that require removal .....                            | 49 |
| Figure 45. Definition of the key metric .....   | 50 |
| Figure 46. Correlation expression and actions for no. of NewOrderInboundEvent instances.....    | 51 |
| Figure 47. Correlation expression and the actions for number of instances for Approval Update.. | 52 |
| Figure 48. End Order Trigger definitions.....   | 53 |
| Figure 49. Definitions for Status Update Trigger.....   | 53 |
| Figure 50. Definition for Country Update trigger .....  | 54 |
| Figure 51. Definition for OrderAmount Update trigger .....                                      | 55 |
| Figure 52. Order Date Timer definition .....  | 57 |
| Figure 53. Order Fulfillment Timer definition .....   | 57 |
| Figure 54. Set the value of the ProcessID key metric field.....                                 | 58 |
| Figure 55. Data Mart model page details for Country and Order Status dimensions .....           | 59 |
| Figure 56. Correct facts for which x is displayed in the Source field .....                     | 59 |
| Figure 57. Example of the corrected measures .....  | 59 |

|  |    |
|--|----|
| Figure 58. Specify Measure and Dimension values .....                                  | 60 |
| Figure 59. Specify correct target and ranges for the KPI .....                         | 61 |
| Figure 60. Add Number of Orders Measure and Order Date Time dimension.....             | 62 |
| Figure 61. Assure Decimal is specified as the Type.....                                | 63 |
| Figure 62. Add Completed string to is Approval Step Completed dimension .....          | 64 |
| Figure 63. Specify Total Order Amount for the Measure .....                            | 65 |
| Figure 64. ID field changed to Total_Sales_Amount_This_Year_for_Canada .....           | 66 |
| Figure 65. Copy .svg file from the artifacts zip file to your project.....             | 67 |
| Figure 66. BPEL model with metrics, counter, and stopwatch.....                        | 69 |
| Figure 67. Add triggers that update metric, counter, and stopwatch .....               | 70 |
| Figure 68. Map for City metric .....   | 71 |
| Figure 69. Current Status metric with Metric Value Maps .....                          | 74 |
| Figure 70. Setting Order Date .....  | 74 |
| Figure 71. Is Approval Step Completed metric with added metric maps for triggers ..... | 75 |
| Figure 72. Ship Percentage metric with its map.....                                    | 76 |
| Figure 73. Add trigger for the Declined Order Counter .....                            | 76 |
| Figure 74. OrderHandlingFuture1 Termination Trigger used to stop the timer .....       | 77 |
| Figure 75. Find this section of the .mm file.....                                      | 77 |
| Figure 76. Data Mart model page details for Country and Order Status dimensions .....  | 79 |
| Figure 77. Process ID, Order Amount, Ship Percentage in fact table.....                | 79 |
| Figure 78. Measures in OrderHandlingFuture1 cube .....                                 | 80 |
| Figure 79. Add value Shipped to Order Status Dimension.....                            | 80 |
| Figure 80. Specify a Name for each range attribute.....                                | 81 |
| Figure 81. Set Measure to Number of Orders and Time dimension to Order Date .....      | 82 |
| Figure 82. Set Type to Decimal .....   | 83 |
| Figure 83. Add Completed string to is Approval Step Completed dimension .....          | 84 |
| Figure 84. Set Measure to Total Order Amount and Time dimension to Order Date.....     | 85 |
| Figure 85. Add the string Canada or US for the Country dimension .....                 | 86 |

## Part 1. Overview

This document describes the two ways of building the Clips and Tacks sample using the monitor model produced from the Business Measures View (BMV) and the monitor model produced from the Business Process Execution Language (BPEL). The dual-model approach uses one model to capture and store the data from the BPEL model. This model sends outbound events to a separate model, the KPI model, which contains all of the key performance indicators (KPIs) that are of interest. The BMV model exported for this sample is created as a KPI model, and so contains essentially only the KPI information. The BPEL model are exported from the BPEL process and essentially contains the generated events and contexts for the model. In the merged model approach, the KPIs from the BMV model are copied into the BPEL model to produce a single model.

This document describes the steps to generate a monitor model based on the Clips and Tacks sample. It does not describe the steps to generate the Clips and Tacks application. However, this application is included in the project interchange file for the sample.

### *Sample project interchange file and enterprise archive files*

To help you generate a monitor model, this sample includes a project interchange (PI) file and enterprise archive (EAR) files.

- **Project interchange file:** ClipsAndTacksSample1.zip contains a complete solution to this sample. It includes these zip files:
  - ClipsAndTackDualModel.zip, which includes the BPEL model and the KPI model
  - ClipsAndTacksMergedModel.zip, which includes a single BPEL model with merged KPI contents.

The project interchange file is ready to be imported into WebSphere Integration Developer.

- **EAR files:** The sample uses these EAR files which are provided with the sample download:
  - ClipsAndTacksApp.ear: the BPEL application
  - ClipsAndTacksHumanCustomEAR.ear and ClipsAndTacksServiceEAR.ear: EAR files for human tasks
  - ClipsAndTacksMergedSampleEAR.ear: a monitor EAR file with a merged KPI model
  - ClipsAndTacksSampleEAR.ear: an application that monitors the BPEL application and sends events to the monitored-KPI application
  - ClipsAndTacksSampleKPIEAR.ear: an application that receives outbound events and sets KPI values

These generated EAR files are ready to be deployed on the Monitor Server.

### *Software requirements*

To run this sample, you need to have installed this software:

- WebSphere Integration Developer v6.0.2 and the WebSphere Business Monitor development toolkit.
- WebSphere Business Monitor v6.0.2 – Basic Installation including the Dashboard Server and Monitor Server
- WebSphere Business Modeler v6.0.2 (optional), to use the KPI model contained in the artifact file

### *KPIs required for the Clip and Tacks application*

From either of the files in the project interchange file (ClipsAndTackDualModel.zip or ClipsAndTacksMergedModel.zip), you can import the Clips and Tacks application into WebSphere Business Modeler. The KPIs are already defined in the imported model.

This section does not describe the exact steps to take using the Business Measures View (BMV). Refer to the Modeler InfoCenter for details on how to use the BMV.

The steps listed here describe at a high level how to create KPIs in the BMV. The model created in the BMV can be exported as an .mm file and imported into WebSphere Integration Developer later. You then update the .mm file using the Monitor tooling in WebSphere Integration Developer. These steps are optional – you can get the .mm file exported from Modeler from the ClipsAndTacksSample1.zip file. The file name is **Order Handling (Future 1) Business Measures.mm**.

The Clips and Tacks application requires these settings for the indicated KPIs:

- Average Order Fulfillment Time KPI:
  - Target type and value set to Duration with target of 3 days
  - Range details use actual value with ranges:
    - One Day, start value of 0 seconds and end value of 1 day
    - Two Days, start value of 1 day and end value of 2 days
    - Three Days, start value of 2 days and end value of 3 days
    - Four Days, start value of 3 days and end value of 4 days
  - The KPI uses a dimension based on the **Order Status** value. (Using the Monitor Model Editor, update the KPI model to check the **Order Status** field for a value of **Shipped**.)

- Percent of Orders Approved This Month KPI:
  - Target type and value set to Number with a target of 90
  - Time period is repeating with a monthly period type based on the period in progress
  - Range details use actual value with ranges:
    - High Range, start value of 90 and end value of 100
    - Low Range, start value of 9 and end value of 90
  - The KPI uses a dimension based on the **is Approval Step Completed** value. (Using the Monitor Model Editor, update the KPI model to check for the **is Approval Step Completed** field for a value of **Completed**.)
- Average Order Price (US Dollars) KPI:
  - Target type and value set to Number and a target value of 300
  - Range details use actual value with ranges:
    - High Range, start value of 300 and end value of 1000
    - Low Range, start value of 0 and end value of 300
- Declined Orders This Month KPI:
  - Target type and value set to Number and a target value of 3
  - Time period is repeating with a monthly period type based on the period in progress
  - Range details use actual value with the ranges:
    - High Range, start value of 3 and end value of 10
    - Low Range, start value of 0 and end value of 3
- Order Count Review KPI:
  - The KPI uses a dimension based on the order status. (Using the Monitor Model Editor, update the KPI model to check the **Order Status** field for a value of **Review Order**.)
- Order Count Cancel KPI:
  - The KPI uses a dimension based on the order status. (Using the Monitor Model Editor, update the KPI model to check the **Order Status** field for a value of **Cancel**.)
- Order Count Cancelled KPI:
  - The KPI uses a dimension based on the Order Status. (Using the Monitor Model Editor, add a check for an **Order Status** field value of **Cancelled**.)
- Order Count Check Account KPI:
  - The KPI uses a dimension based on the Order Status. (Using the Monitor Model Editor, add a check for an **Order Status** field value of **Check Account**.)
- Order Count Ship KPI:
  - The KPI uses a dimension based on the Order Status. (Using the Monitor Model Editor, check for an **Order Status** field value of **Ship Order**.)
- Order Count Shipped KPI:
  - The KPI uses a dimension based on the Order Status. (Using the Monitor Model Editor, add a check for an **Order Status** field value of **Shipped**.)

- Order Count Check Handling Policy KPI:
  - The KPI uses a dimension based on the **Order Status**. (Using the Monitor Model Editor, add a check for an Order Status field value of **Check Order Handling Policy**.)
- Number of Sales This Month KPI:
  - **Target value** and **type** set to Number and target value set to 100
  - **Time period** is repeating with a monthly period type based on the period in progress
  - Range details use actual value with the values:
    - Excellent, start value of 60 and end value of 125
    - Fair, start value of 30 and end value of 60
    - Poor, start value of 0 and end value of 30
- Total Sales Amount This Year KPI:
  - **Target value** and **type** set to Number and target value set to 300000
  - **Time period** is repeating with a yearly period type based on the period in progress
  - Range details use actual value with the values:
    - Poor, start value of 0 and end value of 100000
    - Fair, start value of 100000 and end value of 200000
    - Good, start value of 20000 and end value of 300000
    - Excellent, start value of 300000 and end value of 400000
- Total Sales Amount This Year – Canada KPI:
  - **Target value** and **type** set to Number and target value of 150000
  - **Time period** is repeating with a yearly period type based on the period in progress
  - Range details use actual value with the values:
    - Poor, start value of 0 and end value of 50000
    - Fair, start value of 50000 and end value of 100000
    - Good, start value of 100000 and end value of 150000
    - Excellent, start value of 150000 and end value of 200000
  - The KPI uses a dimension based on the Country. (Using the Monitor Model Editor, add a check for a **Country field** value of **Canada**.)
- Total Sales Amount This Year – US KPI:
  - **Target value** and **type** set to Number and target value of 150000
  - **Timer period** is repeating with a yearly period type based on the period in progress



## 9 Generating MM using BMV and BPEL model

- Range details use actual value with the values:
  - Poor, start value of 0 and end value of 50000
  - Fair, start value of 50000 and end value of 100000
  - Good, start value of 100000 and end value of 150000
  - Excellent, start value of 150000 and end value of 200000
- The KPI uses a dimension based on the Country (Using the Monitor Model Editor, add a check for a **Country field** value of **US**.)

The following figures show the KPIs for the **Average Order Fulfillment Time** metric:

Figure 1. KPIs for the Average Order Fulfillment Time metric – figure 1

The screenshot shows a 'Business Measure Details' dialog box. The title bar reads 'Business Measure Details' with a close button. Below the title bar, the text 'Business Measure Details' and 'Specify additional details to describe how the business measure is calculated.' is displayed. The main area contains several sections:

- Name:** A text box containing 'Average Order Fulfillment Time'.
- Type:** Radio buttons for 'KPI' (selected), 'Instance metric', 'Aggregate metric', and 'Unspecified'.
- Description:** An empty text area.
- Dashboard views:** A section with a right-pointing triangle icon.
- Specify a target value and type:** A checked checkbox. Below it, the text 'The target is an exact value that the KPI should achieve.' is shown. The 'Type' is set to 'Duration' in a dropdown. The 'Target value' is split into four spinners: 'Days' (3), 'Hours' (0), 'Minutes' (0), and 'Seconds' (0).
- Specify a time period over which the business measure will be monitored:** An unchecked checkbox. Below it are three radio buttons: 'Repeating' (selected), 'Rolling', and 'Fixed'. Under 'Repeating', there are three sub-sections:
  - Period type:** A dropdown menu set to 'Yearly'.
  - Size of period:** A spinner set to '90'.
  - Start date:** A date picker set to 'November 02, 20'.
  - End date:** A date picker set to 'December 31, 20'.
  - Base period on:** Radio buttons for 'Last full period' (selected), 'Period in progress', 'Previous day', and 'Day in progress'.

At the bottom right, there are 'OK' and 'Cancel' buttons.

Figure 2. KPIs for the Average Order Fulfillment Time metric – figure 2

**Business Measure Details**

Business Measure Details

Specify additional details to describe how the business measure is calculated.

**Specify range details:**

Ranges can be defined as percentages of the target value or as fixed, actual values.

Percentage of target value (target value = 100%)

Actual value

**Specify ranges**

A range is a set of values, such as allowable margins or lower and upper limits, against which

| Range name | Start value | End value |
|------------|-------------|-----------|
| One Day    | 0 seconds   | < 1 days  |
| Two Days   | 1 days      | < 2 days  |
| Three Days | 2 days      | < 3 days  |
| Four Days  | 3 days      | < 4 days  |

**Specify when to send an alert and the action to take as a result**

For example, when this measure exceeds a certain value, an email may need to be

| Alert description                        |
|--|
| Average Order Fulfillment is over 3 days |

**Specify the dimensions to use to calculate the KPI value**

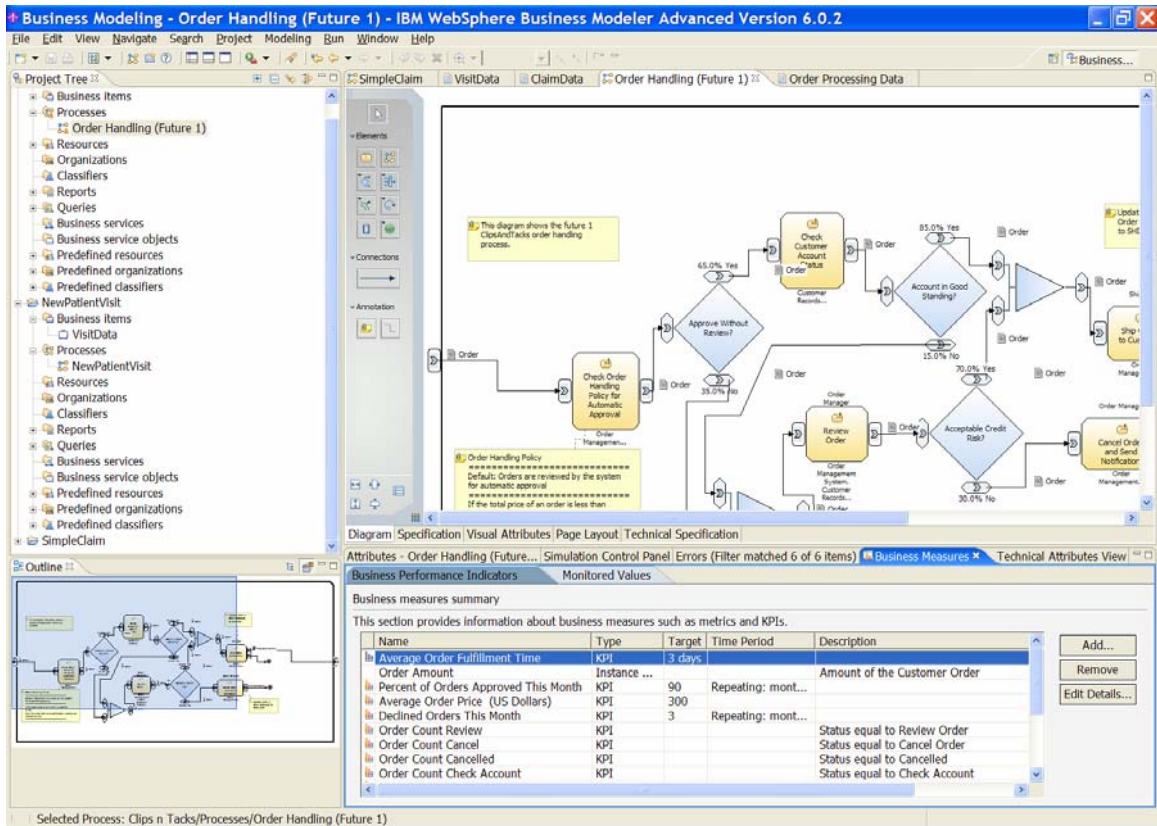
For example: location, sales representative, or date.

| Dimension  |
|--|
| <input type="checkbox"/> Country                 |
| <input checked="" type="checkbox"/> Order Status |

## 11 Generating MM using BMV and BPEL model

The following figure is an example of the process for the Clips and Tacks sample, as shown in the BMV tool. The KPIs have been added to the model.

Figure 3. Clips and Tacks sample, as shown in the BMV tool



## Part 2. Generating monitor models from BMV and the BPEL process

Regardless of whether you use the dual-model approach or the merged-model approach, you first need to generate the monitor models for future use as input.

Instead of following the instructions in this chapter, you can get the artifacts directly from the artifacts files provided with the samples. The instructions show you how to do some aspects of the model generation.

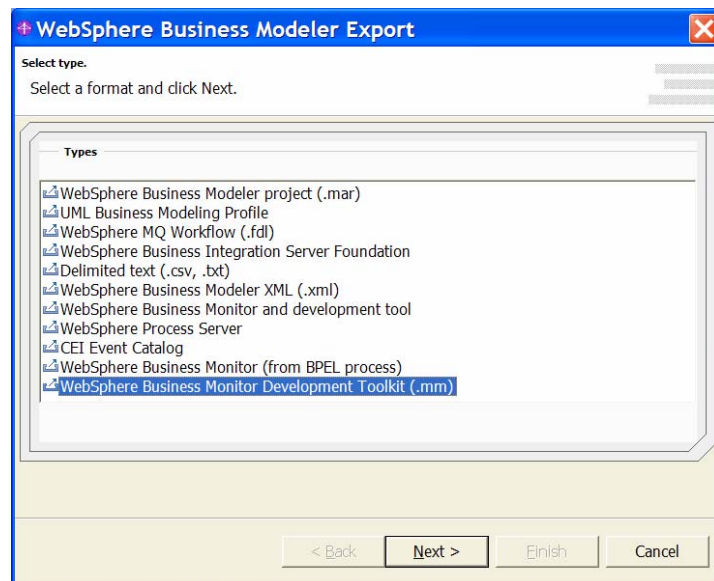
- You can get the .mm file exported from the BMV directly from the **ClipsAndTacksSample1.zip** file. Its name is **Order Handling (Future 1) Business Measures.mm**. Or, if you performed the steps in Part 1, you can export the model from Modeler as an .mm file.
- Import the Clips and Tacks Application project and the related projects into the WebSphere Integration Developer workspace from the **ClipsAndTacksSample1.zip** file.

### *Exporting the monitor model from BMV*

To export the model from Modeler:

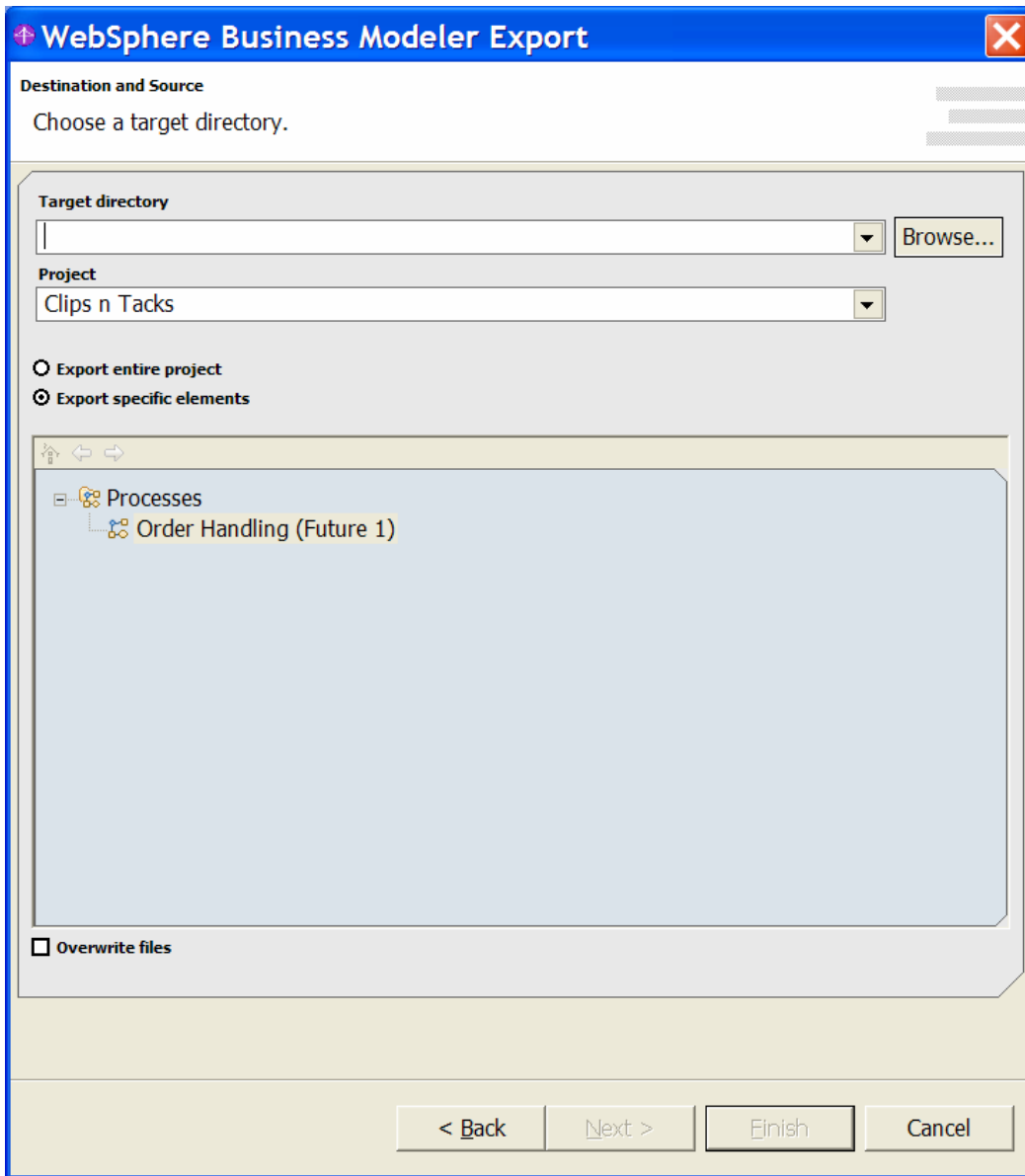
1. Select the process under **Processes** in the project tree and select **Export** from the context menu.
2. In the **Export** window, select the **WebSphere Business Monitor Development Toolkit (.mm)** option.

Figure 4. Select WebSphere Business Monitor Development Toolkit (.mm)



3. Click **Next**. In the **WebSphere Business Modeler Export** window, as shown in this example, enter the target directory, select **Export specific elements**, and select the process to export.

Figure 5. Select Export specific elements and the process to export

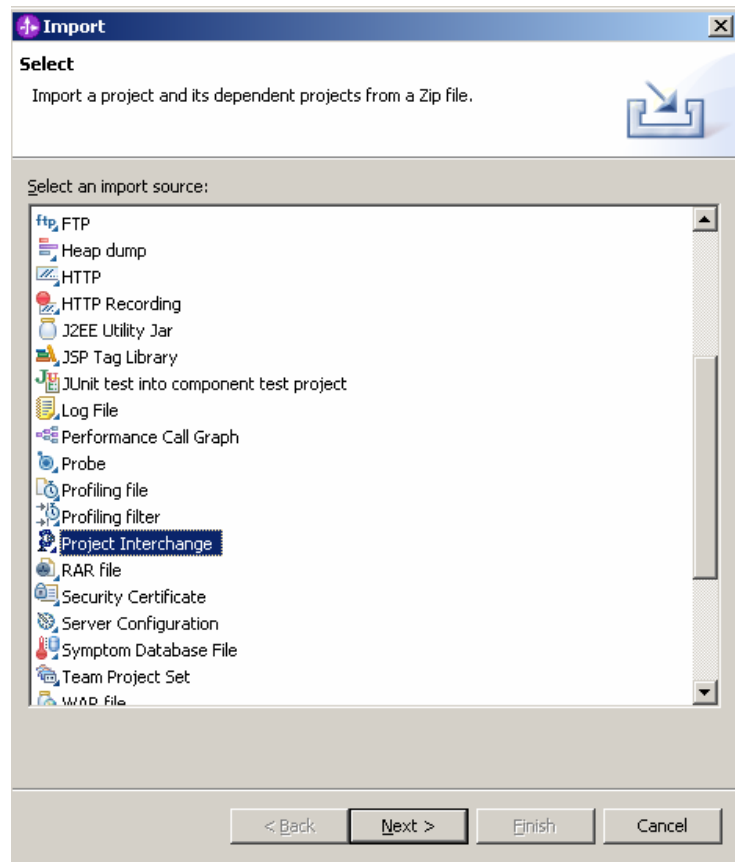


4. Click **Finish** to generate the .mm file.

*Generating the monitor model from the BPEL process*

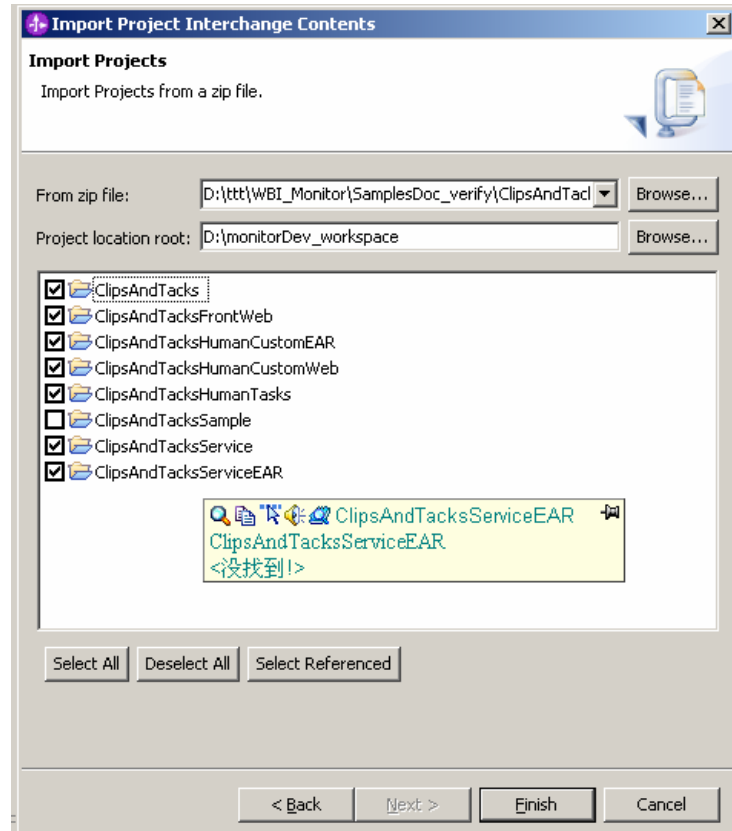
1. Import **BPEL Application** project into **WebSphere Integration Developer**. Unzip **ClipsAndTacksSample1.zip** to a directory, for example, c:\ClipsAndTacksSample1. There are two .zip files unzipped to the directory:
  - ClipsAndTackDualModel.zip
  - ClipsAndTacksMergedModel.zip
2. Start WebSphere Integration Developer and click **File** ► **Import** ► **Project Interchange**, as illustrated below:

Figure 6. Click File ► Import ► Project Interchange



- \_\_\_\_ 3. Click Next and select ClipsAndTackDualModel.zip file to import, Select projects with the exception of **ClipsAndTacksSample** project, as shown in the following figure:

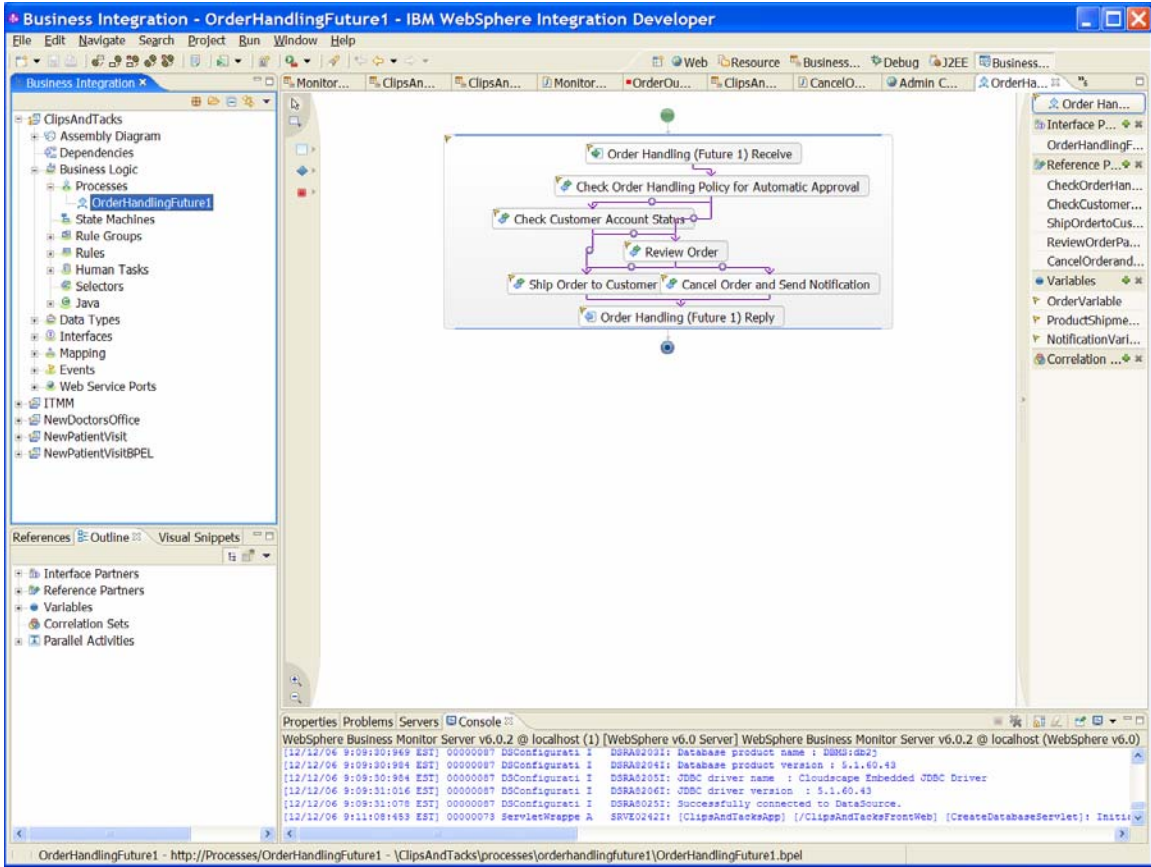
Figure 7. Select projects with the exception of “ClipsAndTacksSample”



- \_\_\_\_ 4. Click **Finish** to finish the project importing.

- \_\_\_\_\_ 5. To generate the monitor model from the BPEL process, select the **OrderHandlingFuture1** process in the **Business Integration** view, as shown in the following example:

Figure 8. Select the OrderHandlingFuture1 process

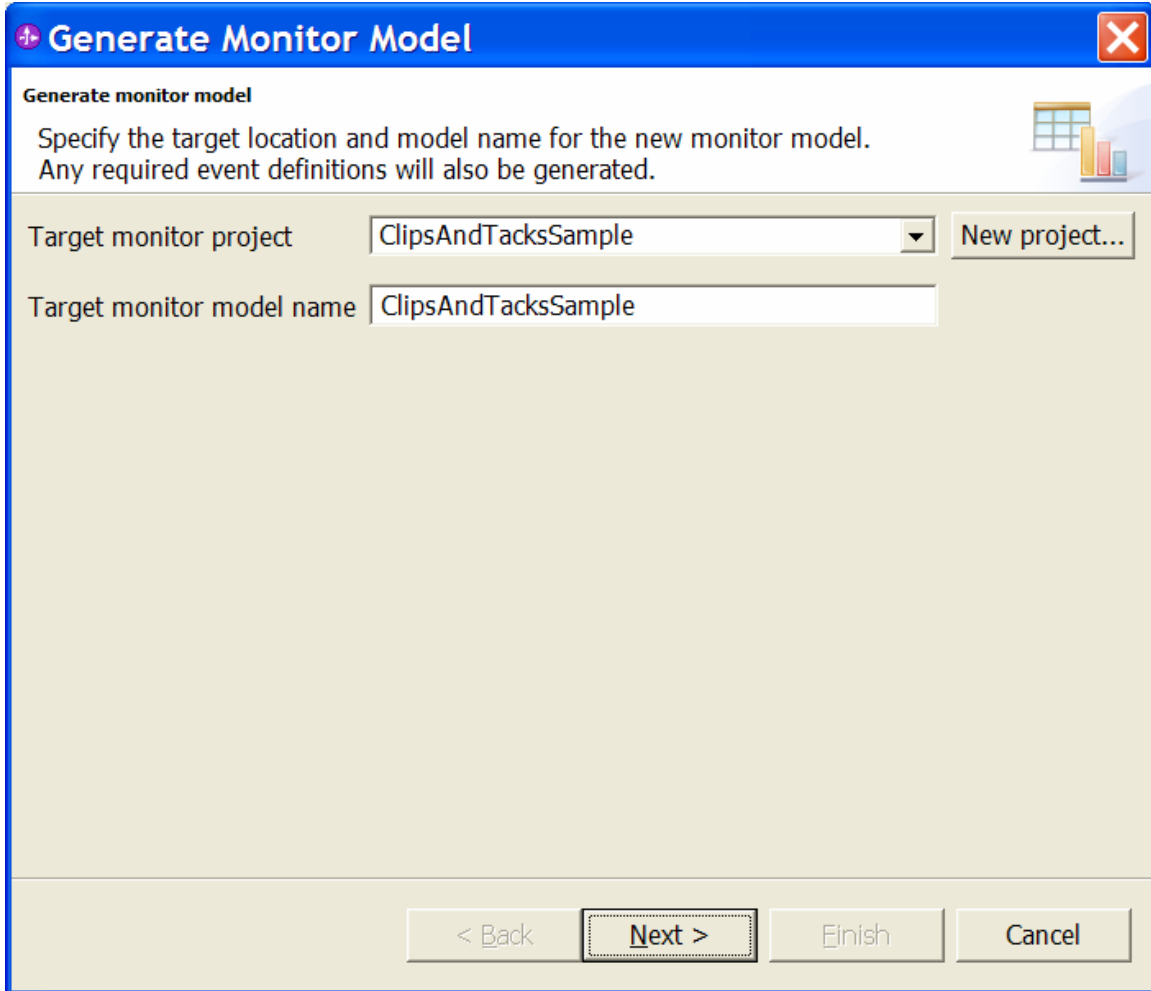


- \_\_\_\_\_ 6. From the context menu for the process, select **Monitor Tools ► Generate Monitor Model**.



- \_\_\_\_ 7. In the next window, either create a new monitor project or select an existing one, and enter the name for the model, as shown in the following example:

Figure 9. Create a new monitor project or select an existing one

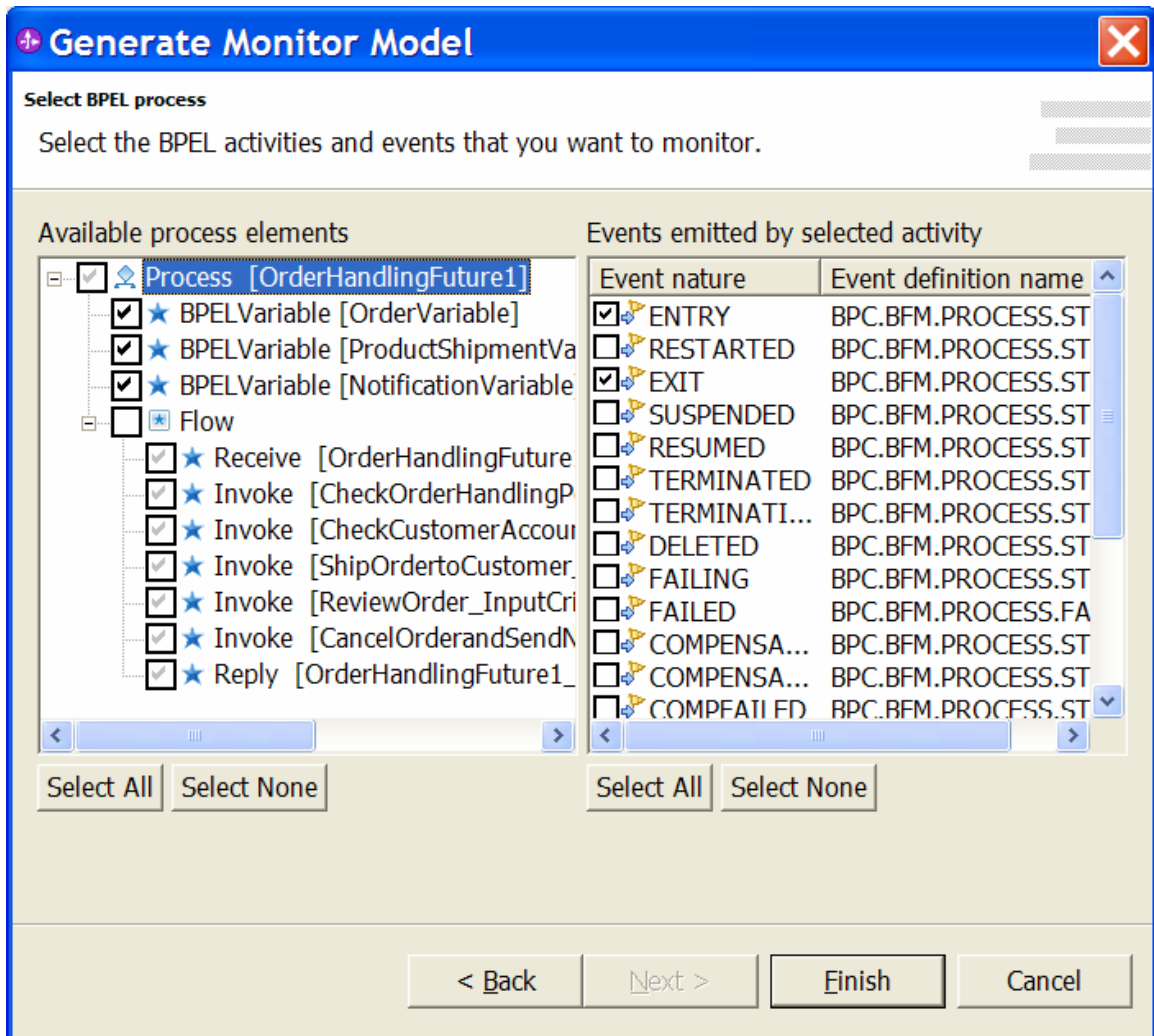


In the next window, select the events that you are interested in.

- **Main process:** Entry and Exit events
- **Activities:** Entry and Exit events for the activities
  - The activities are listed under the **Flow** item.
  - Only the Exit event is available for the **Reply** activity, so select that.
- **BPEL Variable process elements:** Changed event

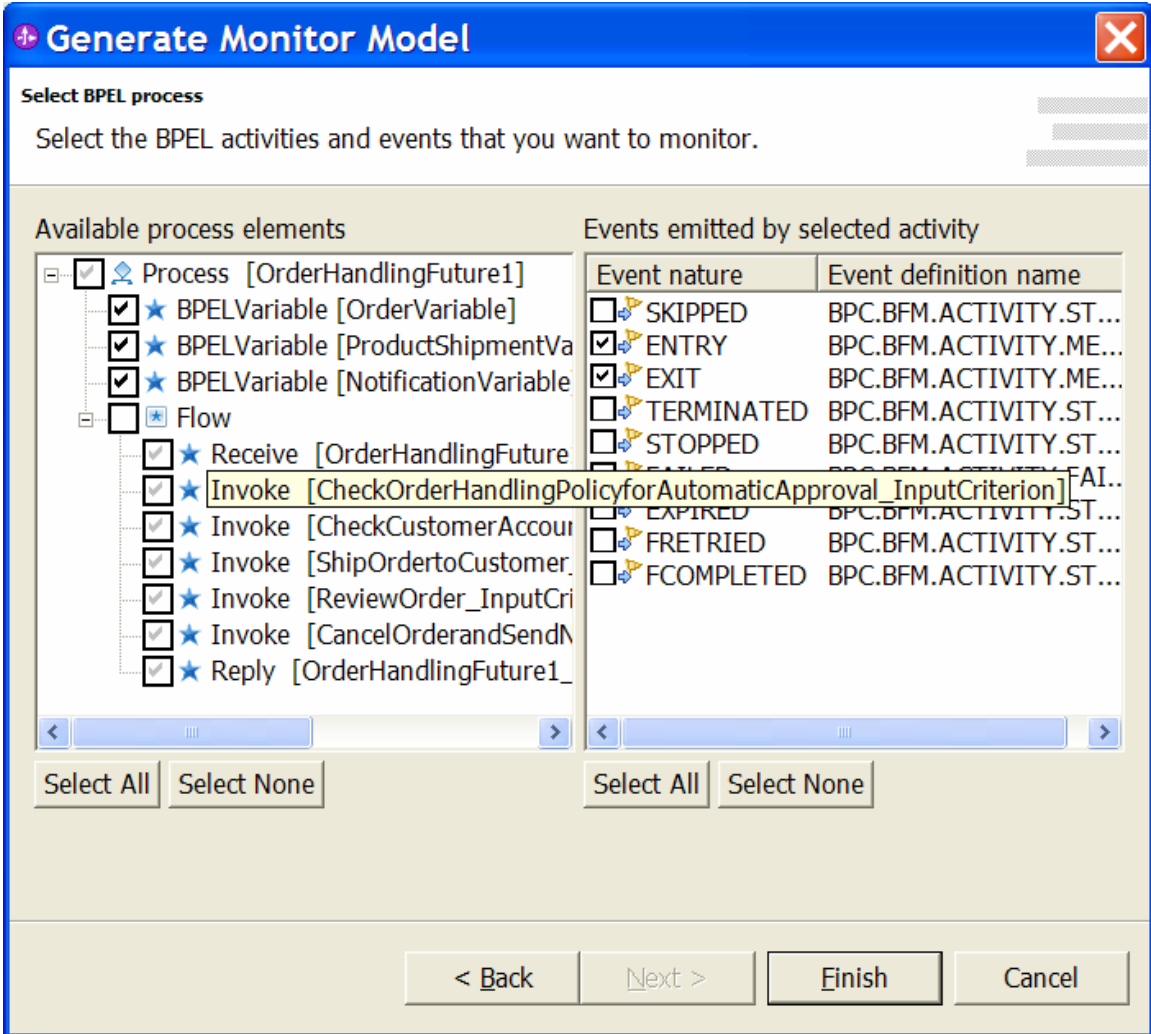
The following examples illustrate the selections for the process, one of the activities, and one of the BPELVariable entries:

Figure 10. Process selection Entry and Exit events



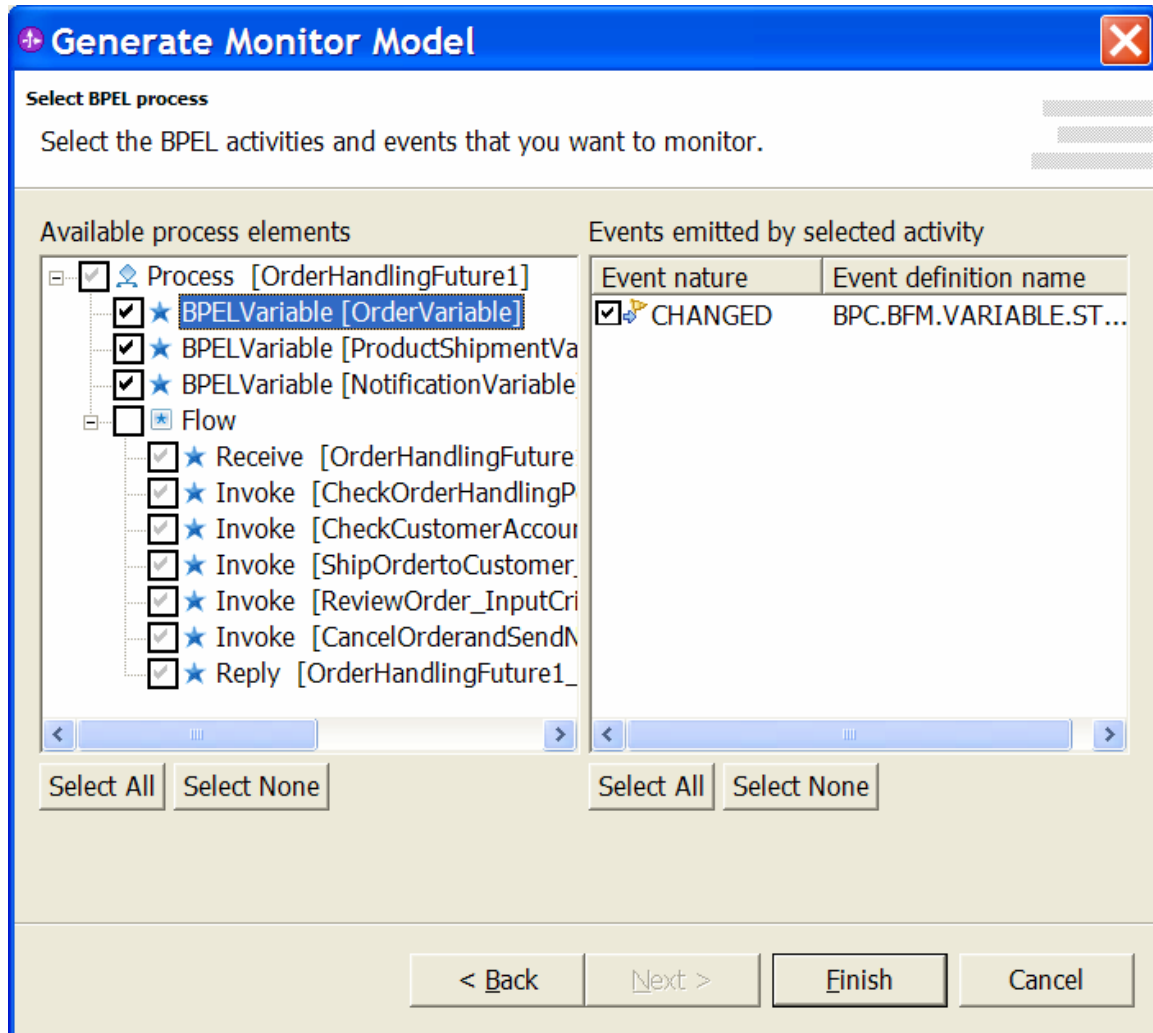
The following example illustrates Entry and Exit events for the activities:

Figure 11. Entry and Exit events for the activities



The following example illustrates BPELVariable entries:

Figure 12. BPELVariable entries



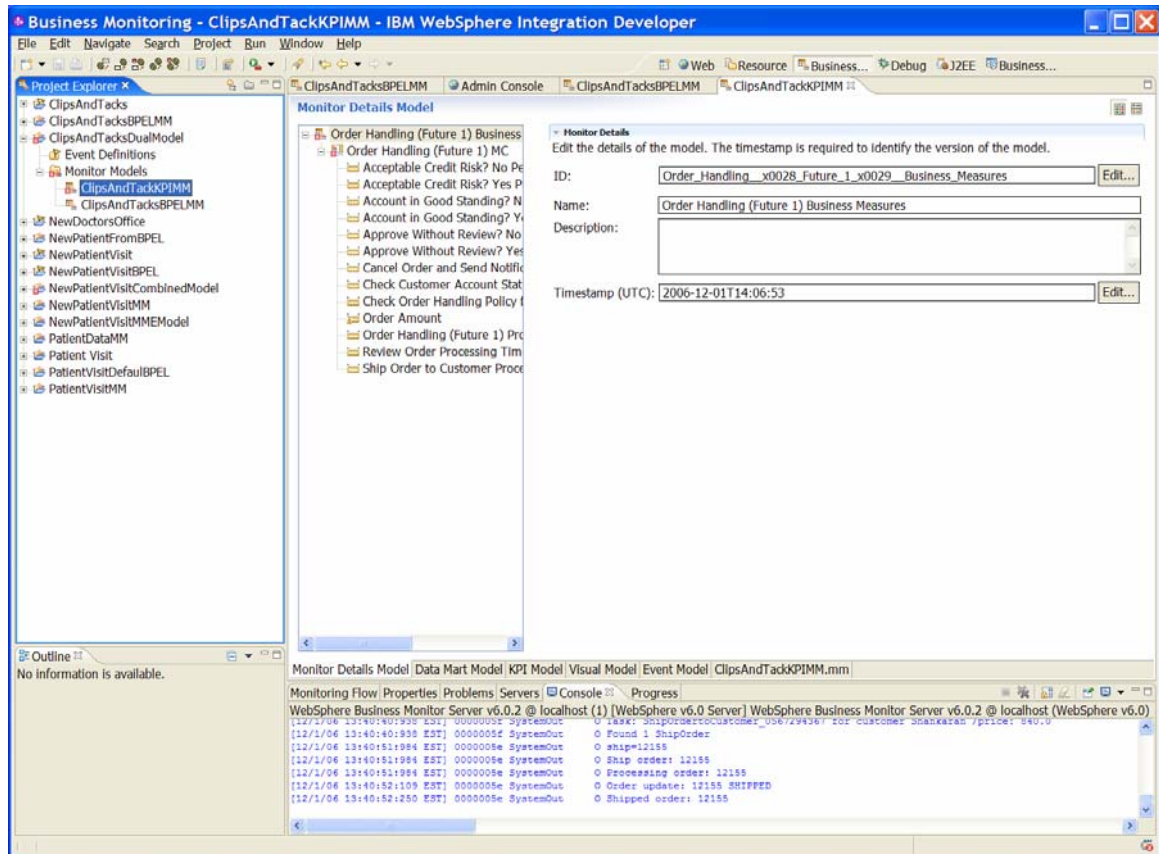
\_\_\_\_ 8. Click **Finish** to create the events and the monitor model.

### Part 3. Using the dual-model approach

For the dual-model approach, the monitor model generated from BMV and the model generated from the BPEL are placed in the same Business Monitoring project. Assuming the Monitor project is created from the BPEL Export, use **File ► System Import** to import the BMV files into the same project that contains the BPEL Monitor model and change the name of the BMV model to ClipsAndTackKPIMM.

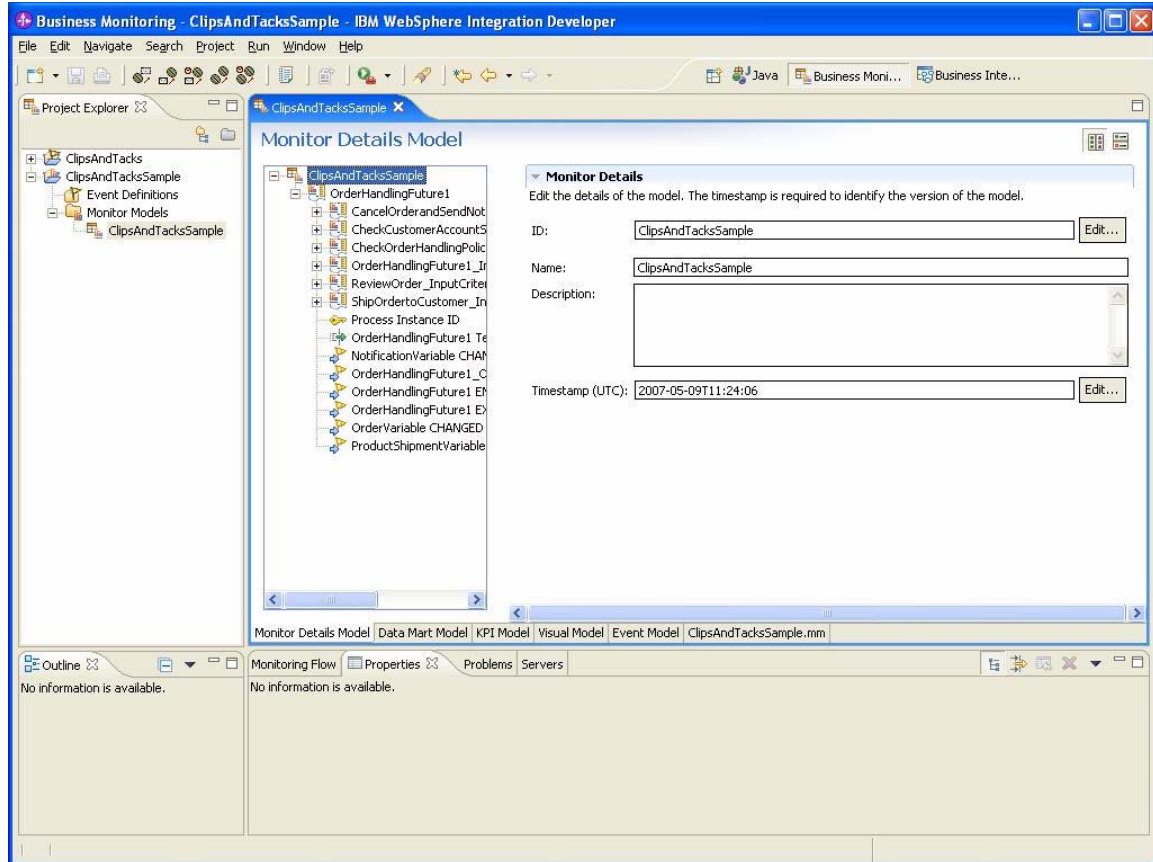
The following figure illustrates the KPI model (ClipsAndTackKPIMM) that is generated from BMV:

Figure 13. KPI model (ClipsAndTackKPIMM) generated from BMV



Here is the BPEL-generated model (ClipsAndTacksSample):

Figure 14. BPEL-generated model (ClipsAndTacksSample)



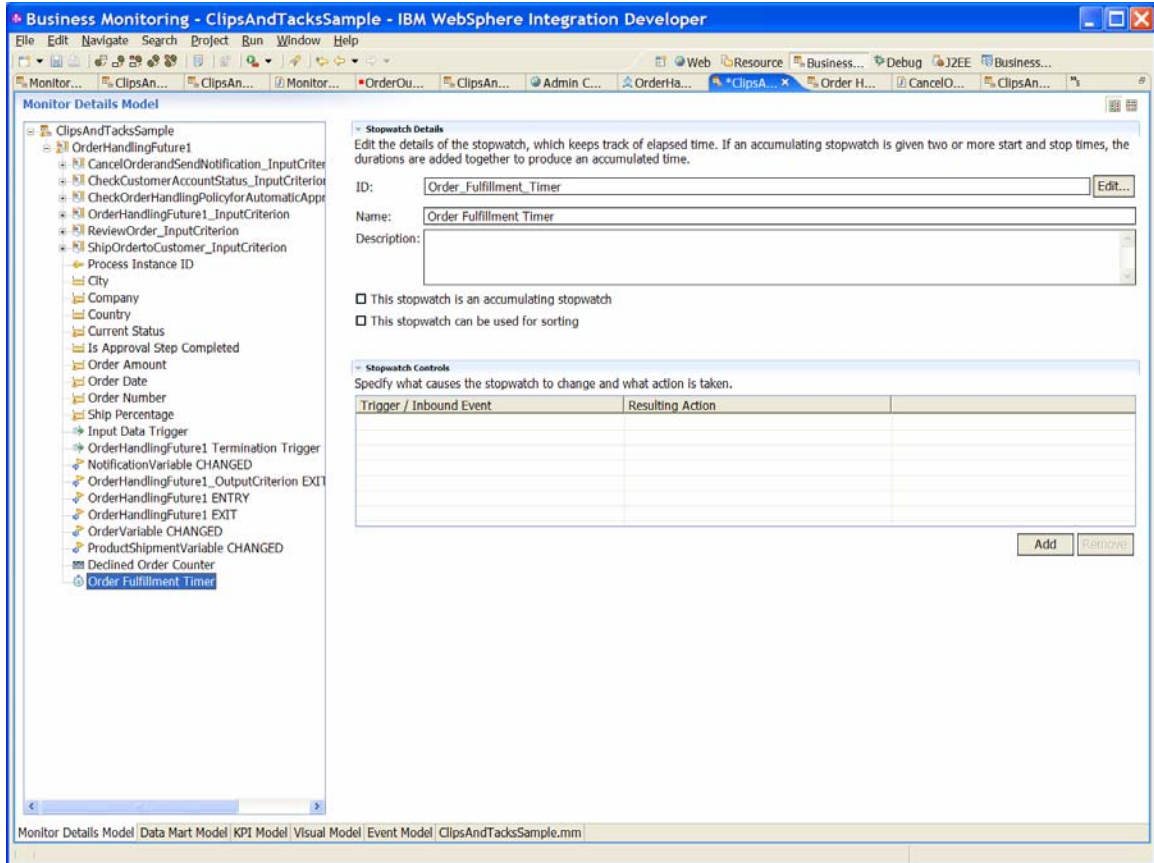
#### *Defining additional data needed by the KPI model*

1. First, add all of the metrics, counters, and stopwatches to the BPEL model to capture the data needed for the KPI model. Specifically, add these metrics to the OrderHandlingFuture1 monitor context:
  - **City:** string, maximum length of 256, default value of Durham
  - **Company:** string, maximum length of 256, no default value
  - **Country:** string, maximum length of 256, default value of US
  - **Current Status:** string, maximum length of 256, no default value
  - **Is Approval Step Completed:** string, maximum length of 256, default value of Not Completed
  - **Order Amount:** decimal, no default value
  - **Order Date:** date, no default value
  - **Order Number:** integer, no default value
  - **Ship Percentage:** integer, default value of 0

- \_\_\_\_\_ 2. Next, add the **Declined Order Counter** and the **Order Fulfillment Timer** to the OrderHandlingFuture1 monitor context.

Here is the BPEL model with the metrics, counter, and stopwatch added:

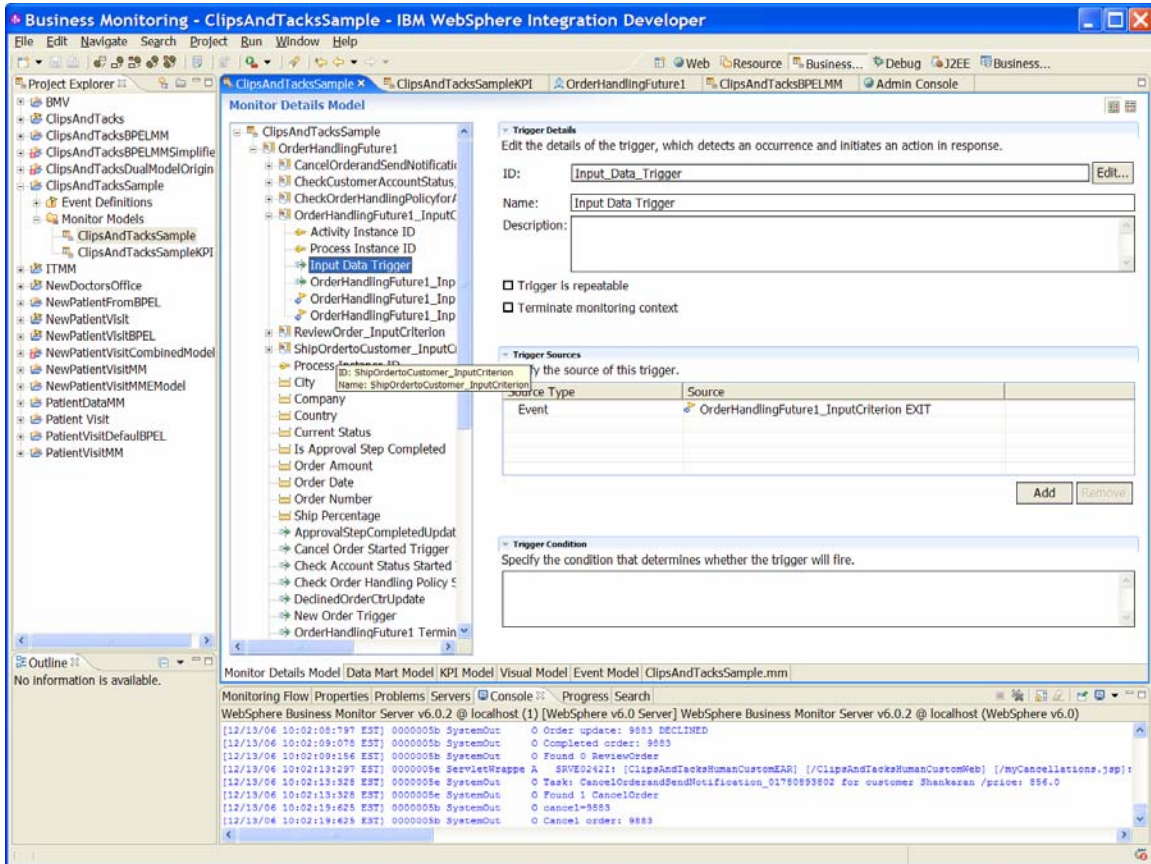
Figure 15. BPEL model with the metrics, counter, and stopwatch



- \_\_\_\_\_ 3. Next, add the triggers that cause the metrics, counters, and stopwatches to get updated. First, add a trigger that is associated with the OrderHandlingFuture1\_InputCriterion EXIT inbound event. This inbound event contains the data entered in the BPEL process. Use this trigger later to set the value in City, Company, Country, Order Amount, and Order Number. So first create the trigger, Input Data Trigger, and add a trigger source for the OrderHandlingFuture1\_InputCriterion EXIT event.

The new trigger is added to the OrderHandlingFuture1\_InputCriterion context. Here is an example of the new trigger:

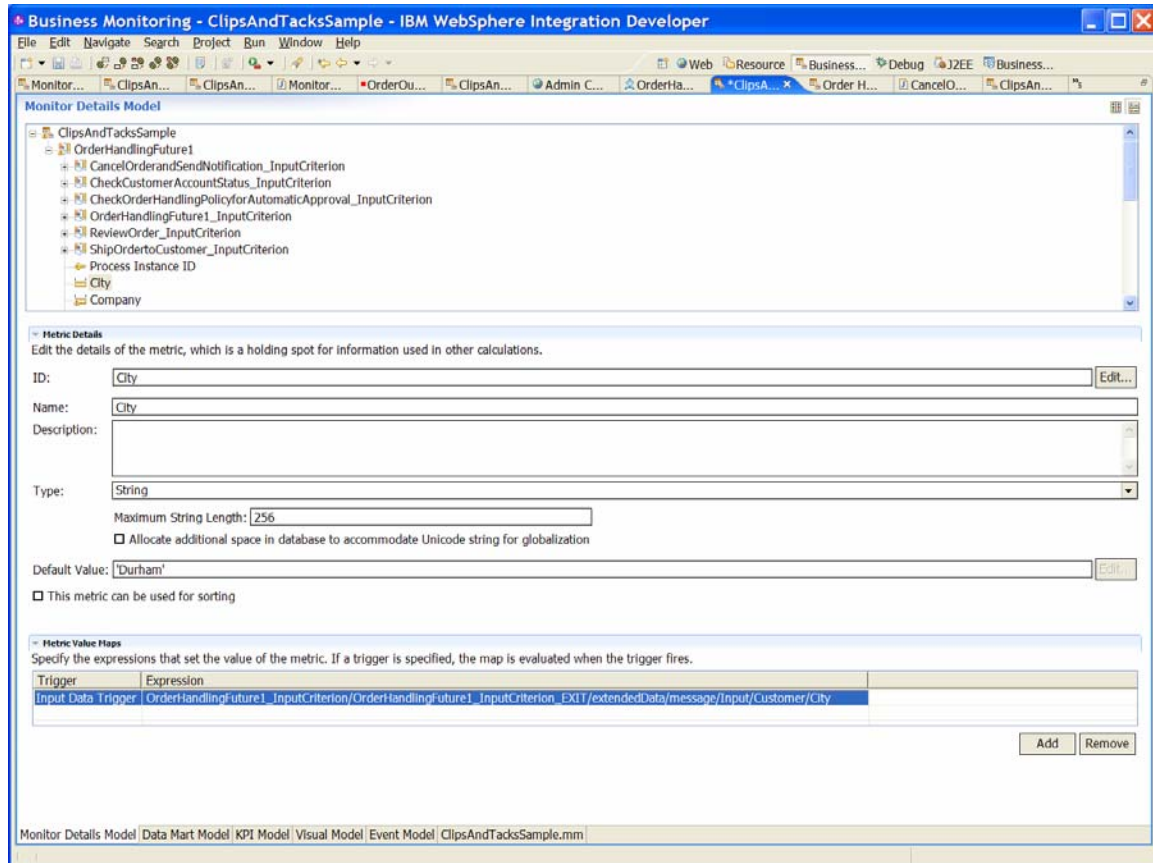
Figure 16. Example of the new trigger





- \_\_\_\_\_ 4. Next, for the metrics to be updated when this trigger fires, add a map on the associated metric that sets the metric from the value in the inbound event. Here is the map for the City metric displayed with vertical orientation so that the entire map expression can be displayed:

Figure 17. Map for the City metric displayed with vertical orientation



- \_\_\_\_\_ 5. Add the maps for the Company, Country, Order Amount, and Order Number metrics that get set by this trigger. The Order Amount is set to the Total Price field from the inbound event.
- \_\_\_\_\_ 6. Next, add the triggers that set the Current Status metric. The Current Status metric is set to one these values (depending upon the location of the order in the order process): Started, Cancelled, Shipped, Cancel Order, Check Account, Checked Account, Review Order, Reviewed, Check Order Handling Policy, Checked Order Handling Policy, and Ship Order.

The triggers listed here, in no particular order, are used to set the status. The termination triggers are generated for you in the BPEL model, but you need to add the other triggers.

For triggers that need to be added, this list identifies the monitor context where the trigger is to be added, and the event source for the trigger.

- New Order Trigger
  - **Monitor context:** OrderHandling Future1
  - **Event source:** OrderHandlingFuture1 ENTRY inbound event
  - Needs to be added
- CancelOrderandSendNotification\_InputCriterion Termination Trigger
  - **Monitor context:** CancelOrderandSendNotification\_InputCriterion
  - Should already exist in the model, because termination triggers are automatically created in the default BPEL model
- ShipOrdertoCustomer\_InputCriterion Termination Trigger
  - **Monitor context:** ShipOrdertoCustomerInputCriterion
  - Should already exist
- Cancel Order Started Trigger
  - **Monitor context:** CancelOrderandSendNotification\_InputCriterion MC
  - **Event source:** CancelOrderandSendNotification\_InputCriterion ENTRY inbound event.
  - This trigger needs to be created. The specified inbound event should be in the generated model.
- Check Account Status Started Trigger
  - **Monitor context:** CheckCustomerAccountStatus\_InputCriterion MC
  - **Event source:** CheckCustomerAccountStatus\_InputCriterion ENTRY inbound event
  - This trigger needs to be created. The specified inbound event should already exist in the specified model.
- CheckCustomerAccountStatus\_InputCriterion Termination Trigger
  - **Monitor context:** CheckCustomerAccountStatus\_InputCriterion MC.
  - This trigger should already exist.
- Review Order Started Trigger
  - **Monitor context:** ReviewOrder\_InputCriterion MC
  - **Event source:** ReviewOrder\_InputCriterion ENTRY inbound event
  - This trigger needs to be created. The specified inbound event should already exist in the generated model.

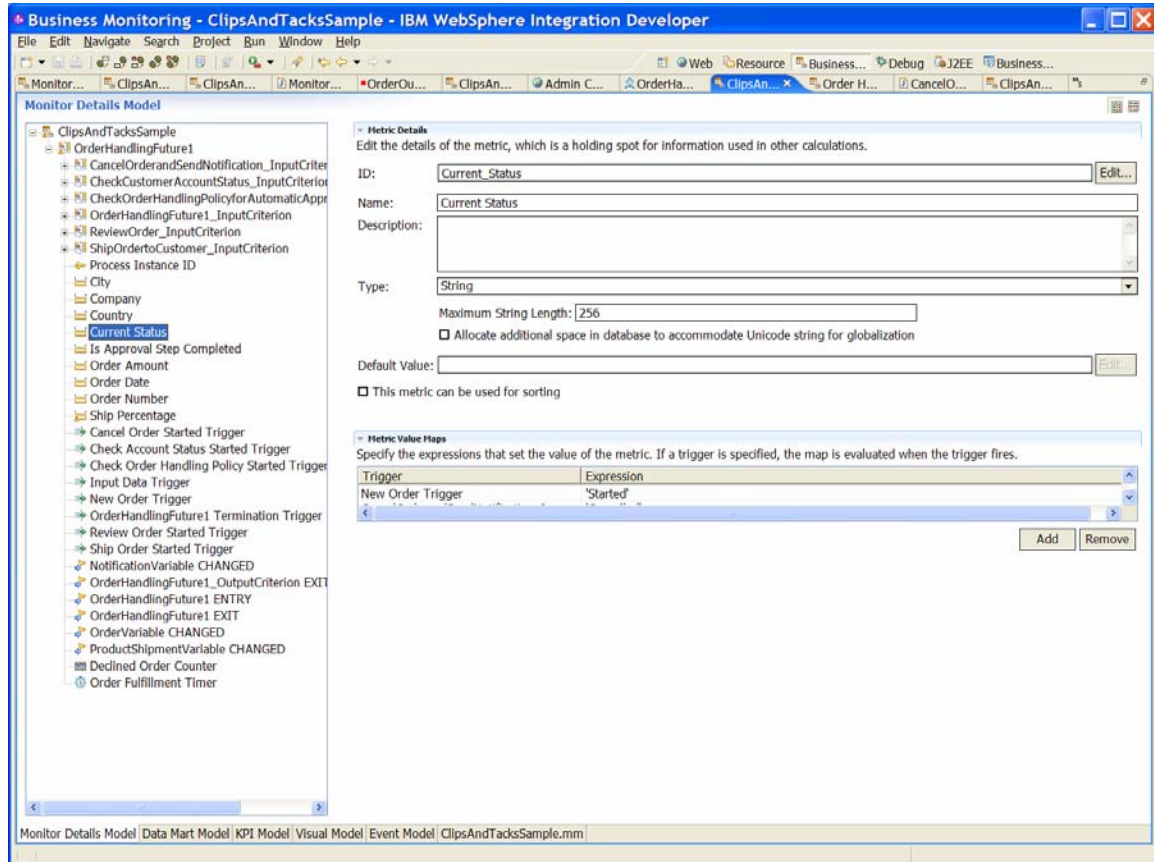
- Review Order\_InputCriterion Termination Trigger
  - **Monitor context:** ReviewOrder\_InputCriterion MC
  - This trigger should already exist
- Check Order Handling Policy Started Trigger
  - **Monitor context:** in the CheckOrderHandlingPolicyforAutomaticApproval\_InputCriterion MC
  - **Event source:** CheckOrderHandlingPolicyforAutomaticApproval\_InputCriterion ENTRY inbound event.
  - This trigger needs to be created. The specified inbound event should already exist in the generated model.
- CheckOrderHandlingPolicyforAutomaticApproval\_InputCriterion Termination Trigger
  - **Monitor context:** CheckOrderHandlingPolicyforAutomaticApproval\_InputCriterion MC.
  - This trigger should already exist.
- Ship Order Started Trigger
  - **Monitor context:** ShipOrdertoCustomer\_InputCriterion MC
  - **Event source:** ShipOrdertoCustomer\_InputCriterion ENTRY event
  - This trigger needs to be created.

\_\_\_\_ 7. After adding the new triggers, add the maps to the Current Status metric to set the correct value for each trigger. For each of the following triggers, specify the indicated value as the setting for the Current Status metric when that trigger fires:

- New Order Trigger – Started
- CancelOrderandSendNotification\_InputCriterion Termination Trigger – Cancelled
- ShipOrdertoCustomer\_InputCriterion Termination Trigger – Shipped
- Cancel Order Started Trigger – Cancel Order
- Check Account Status Started Trigger – Check Account
- CheckCustomerAccountStatus\_InputCriterion Termination Trigger – Checked Account
- Review Order Started Trigger – Review Order
- ReviewOrder\_InputCriterion Termination Trigger – Reviewed
- Check Order Handling Policy Started Trigger – Check Order Handling Policy
- CheckOrderHandlingPolicyforAutomaticApproval\_InputCriterion Termination Trigger – Checked Order Handling Policy
- Ship Order Started Trigger – Ship Order

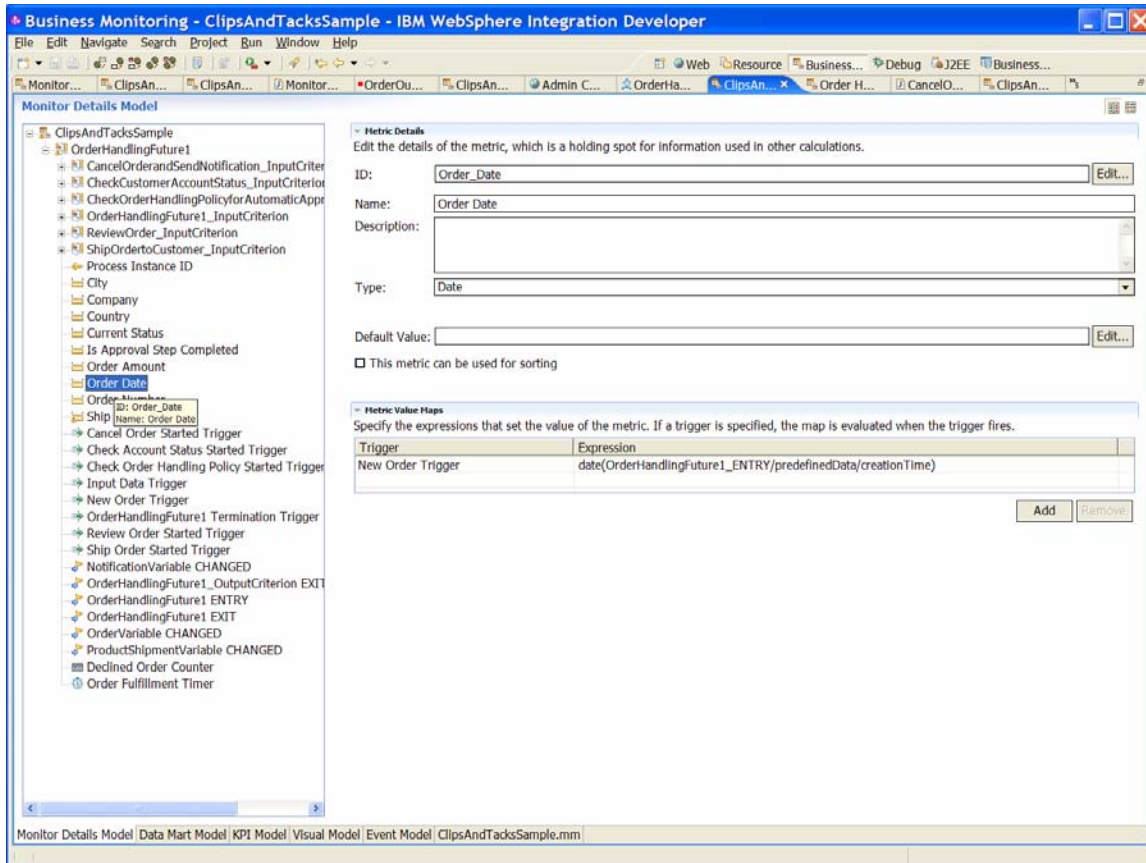
Here is an example of the **Current Status** metric with the **Metric Value Maps** values displayed for **New Order Trigger**:

Figure 18. Metric Value Maps values displayed for New Order Trigger



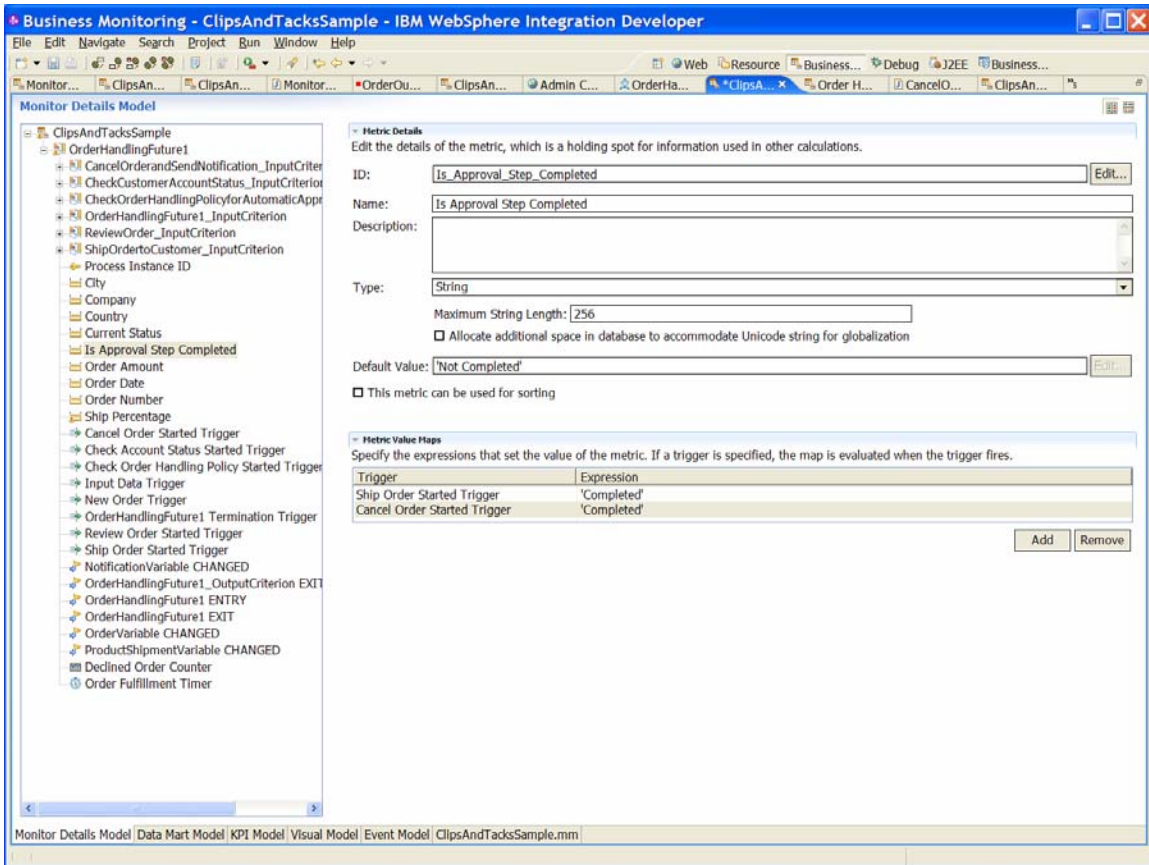
8. Now define the map for the **Order Date** metric. Use the date function with the creationTime field in the **OrderHandlingFuture1** ENTRY inbound event, as shown in this example. (The creationTime is a dateTime field whose type must be converted to **Date** before it can be placed in the **Order Date** metric, whose type is **Date**.)

Figure 19. Define the map for the Order Date metric



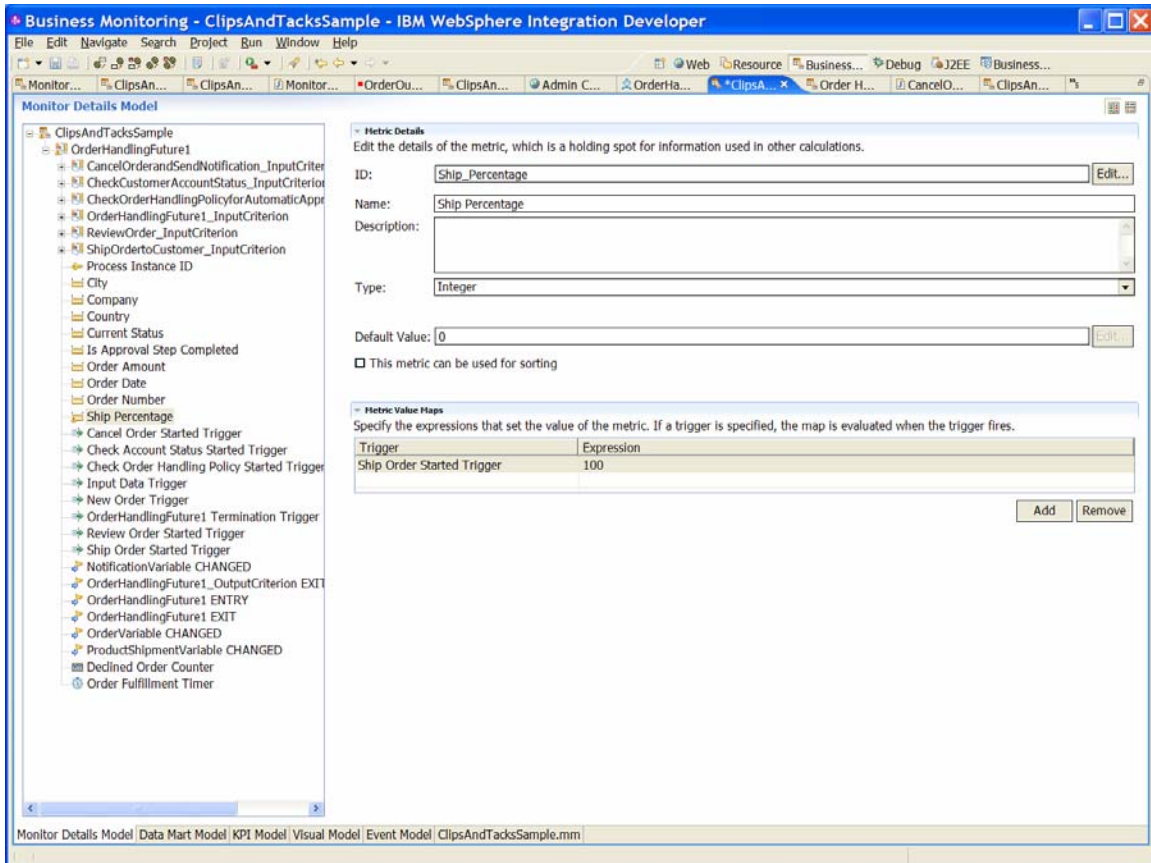
9. The **Is Approval Step Completed** metric is also set by triggers, so you need to add the maps to set this value next. This metric is set to **Completed** when either **Ship Order Started Trigger** or **Cancel Order Started Trigger** is fired. Here is an example of this metric with the added metric maps for the triggers:

Figure 20. Metric with the added metric maps for the triggers



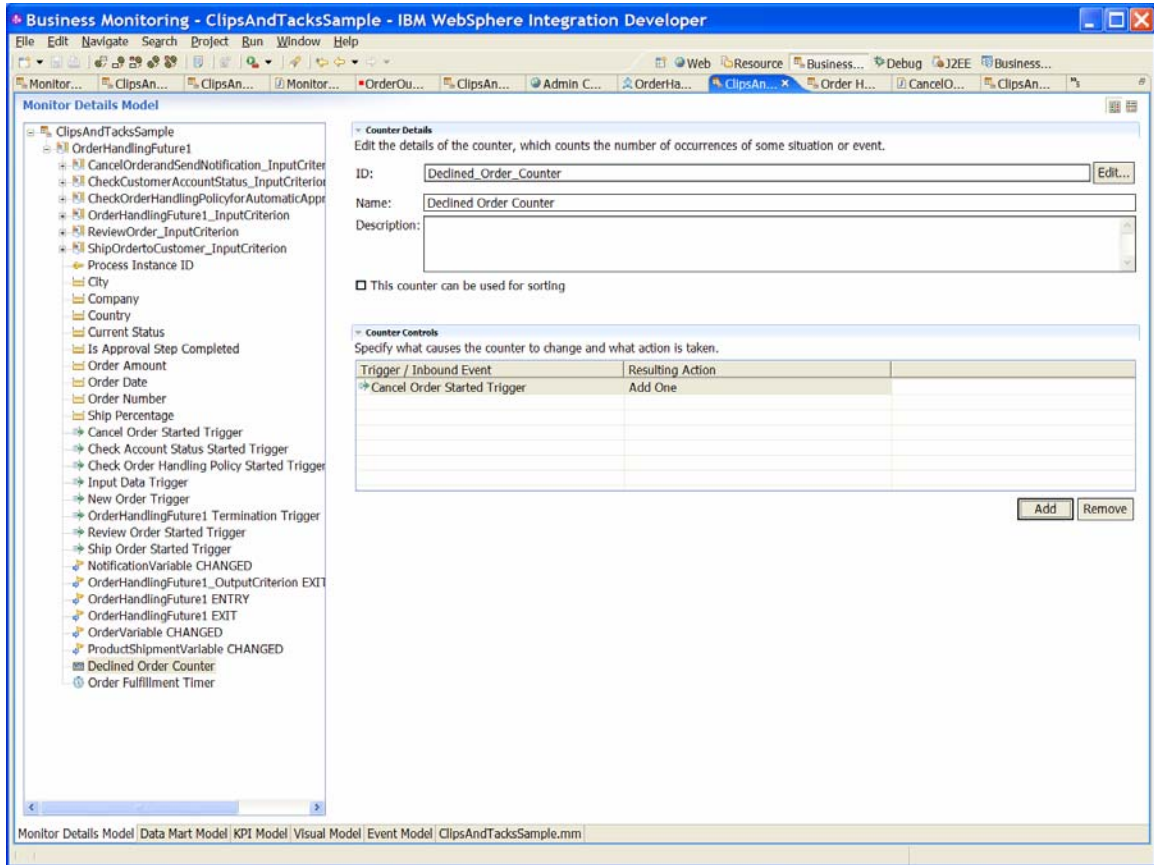
\_\_\_\_ 10. The **Ship Percentage** metric is set to 100 when the **Ship Order Started Trigger** is fired. Shown here is the **Ship Percentage** metric with its map:

Figure 21. Ship Percentage metric with its map



- \_\_\_\_\_ 11. Next, add the trigger for the **Declined Order Counter**.  
Whenever the **Cancel Order Started Trigger** is fired, the **Declined Order Counter** trigger should be incremented.  
This is show here:

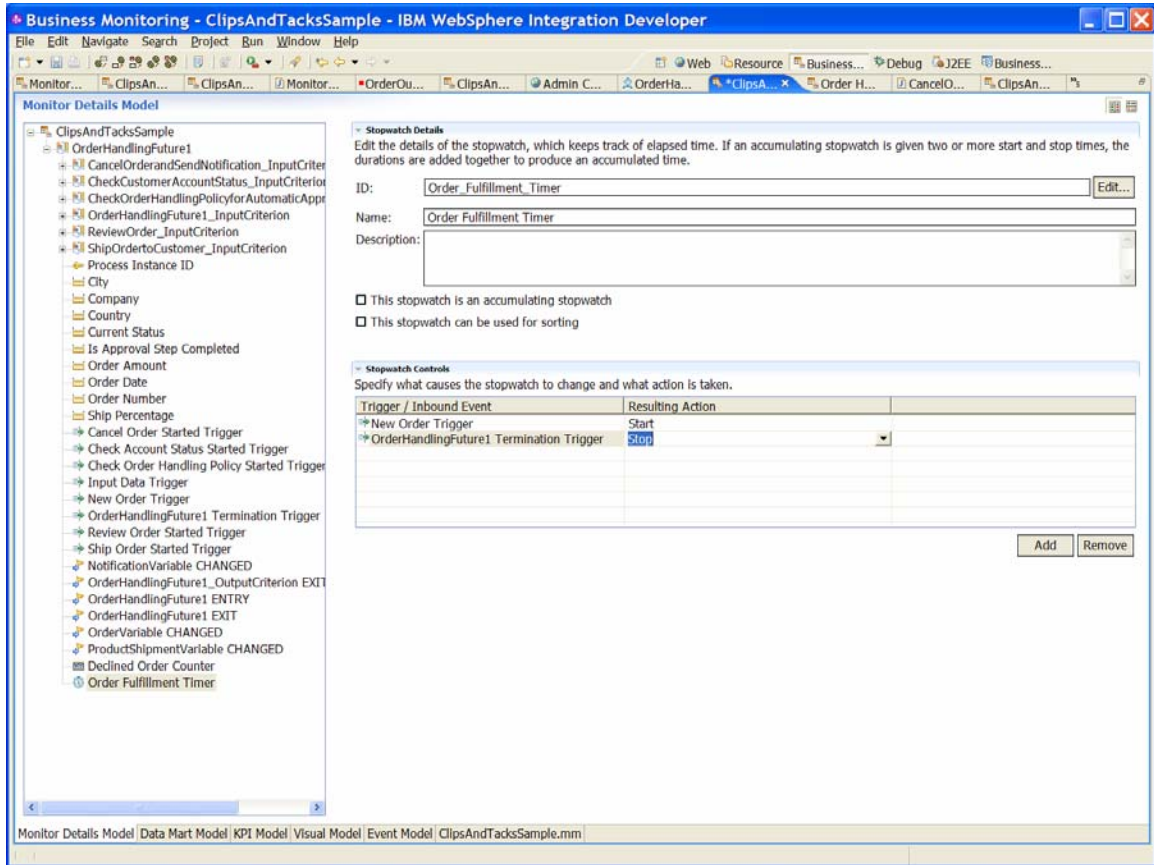
Figure 22. Declined Order Counter trigger counter





- \_\_\_\_\_ 12. Next, add the triggers that start and stop the **Order Fulfillment Timer**. This timer is started when the order is received, so the **New Order Trigger** starts the timer. The timer is stopped when the process is terminated, so the **OrderHandlingFuture1 Termination Trigger** is used to stop the timer. Here is an example of setting these triggers:

Figure 23. OrderHandlingFuture1 Termination Trigger settings



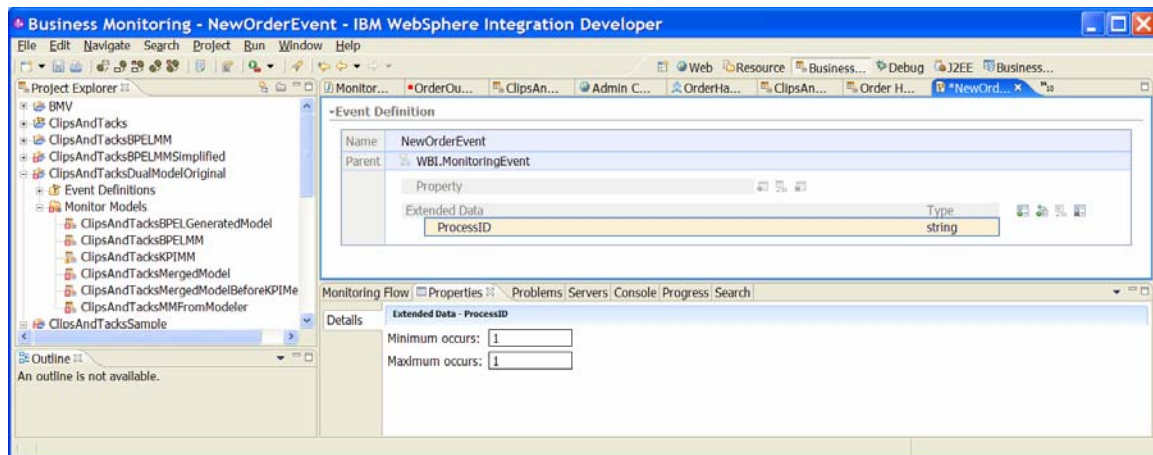
*Defining additional events to be sent to the KPI model*

- \_\_\_\_\_ 1. After you have defined the data that is needed for the KPI model, you need to define the events that are needed to populate the KPI model. These events are used to send data to the KPI model:
- **NewOrderEvent**: sent when the New Order Trigger is fired. Contains the ProcessID value from the BPEL model that are used as the key for the KPI model.
  - **ApprovalStepUpdate**: sent whenever the isApprovalStepCompleted metric is updated. Contains the isApprovalStepCompleted and ProcessID values.
  - **CompleteOrder**: sent when the OrderHandlingFuture1 Termination Trigger is fired and indicates the end of processing for this order. Contains only the ProcessID value.

- **DeclinedOrderCtrUpdate:** sent whenever the Declined Order Counter is updated. Contains the ProcessID and Declined Order Counter values.
- **ReceiveOrder:** sent when the Received\_Order\_Variable\_Changed\_Trigger is fired, and provides the order information to the KPI model. Contains the ProcessID, Country, Order Status, Approval Step Completed, Order Amount, Order Date, Ship Percentage, and the Declined Order Counter values.
- **ShipPercentageUpdate:** sent whenever the Ship Percentage value changes. Contains the ProcessID and the Ship Percentage values.
- **StatusUpdate:** sent whenever the Current Status metric has a value change. This event contains the ProcessID and the Current Status.
- **CountryUpdate:** sent whenever the Country metric has a value change. This event contains the ProcessID and the Country.
- **OrderAmountUpdate:** sent whenever the Order Amount metric has a value change. This event contains the ProcessID and the OrderAmount.

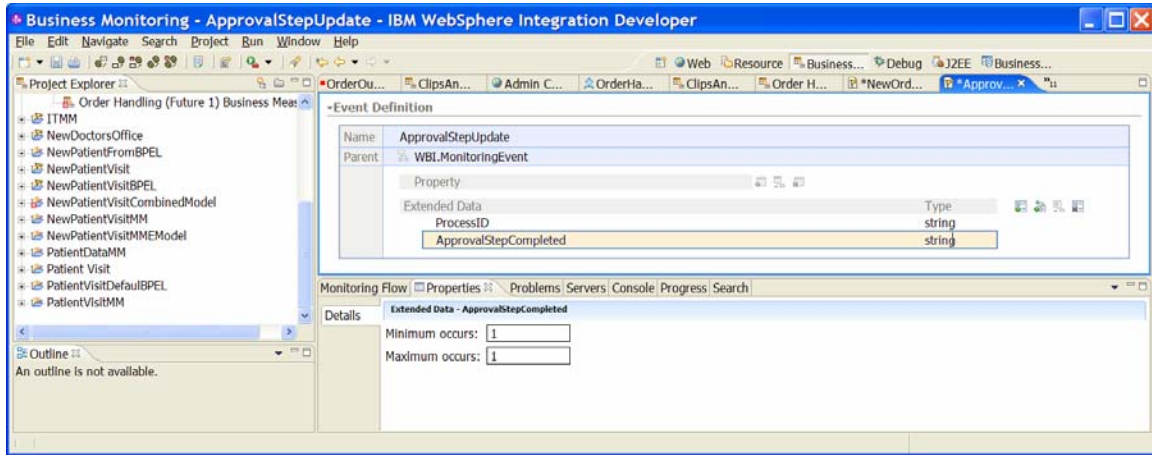
The following figure shows the definition of each of these events:

Figure 24. Definition of the NewOrderEvent



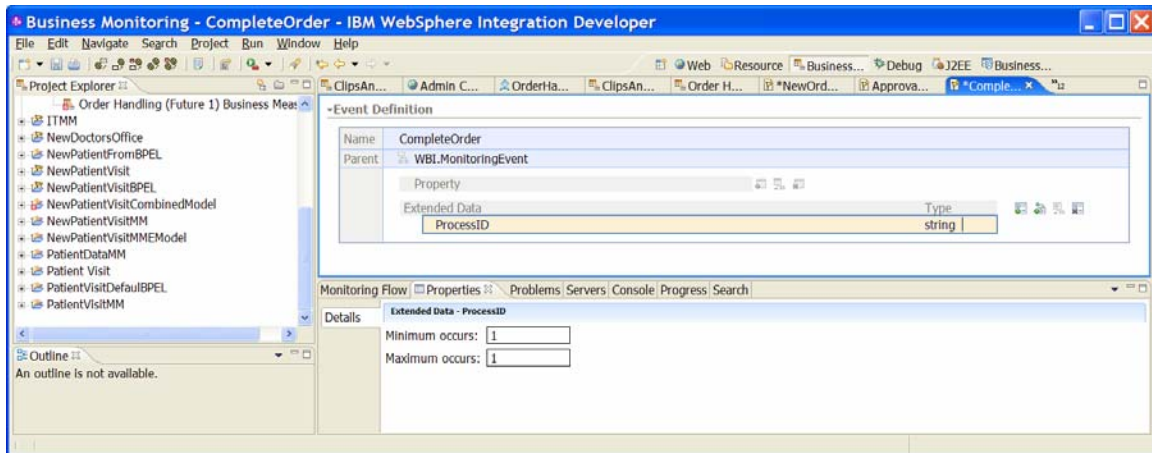
Definition of the **ApprovalStepUpdate**:

Figure 25. Definition of the ApprovalStepUpdate



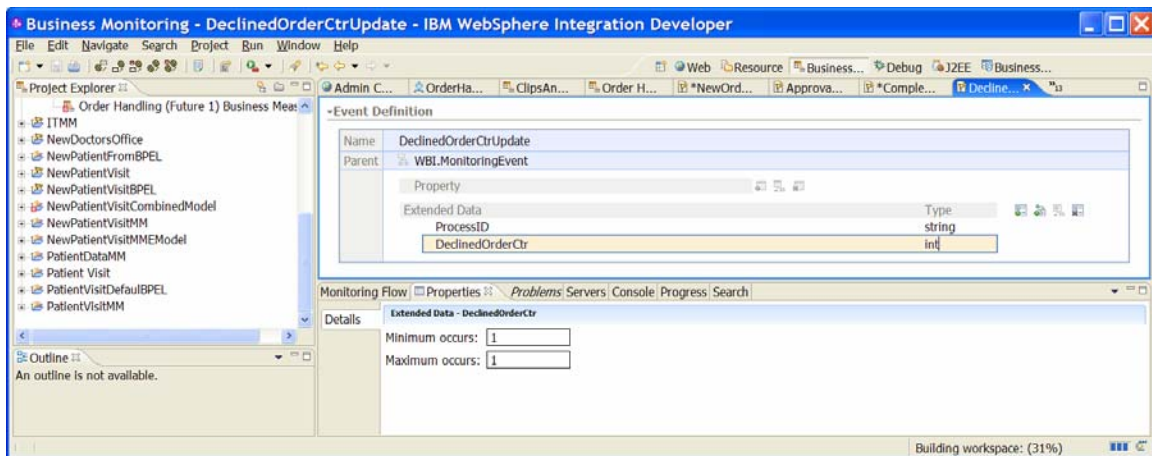
Definition of the **CompleteOrder**:

Figure 26. Definition of the CompleteOrder



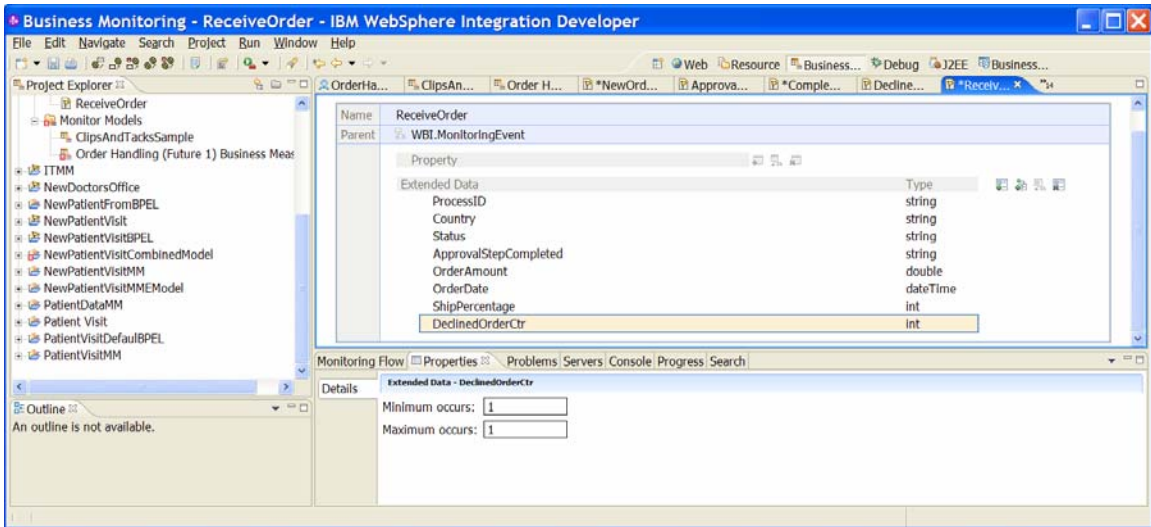
Definition of the **DeclinedOrderCtrUpdate**:

Figure 27. Definition of the DeclinedOrderCtrUpdate



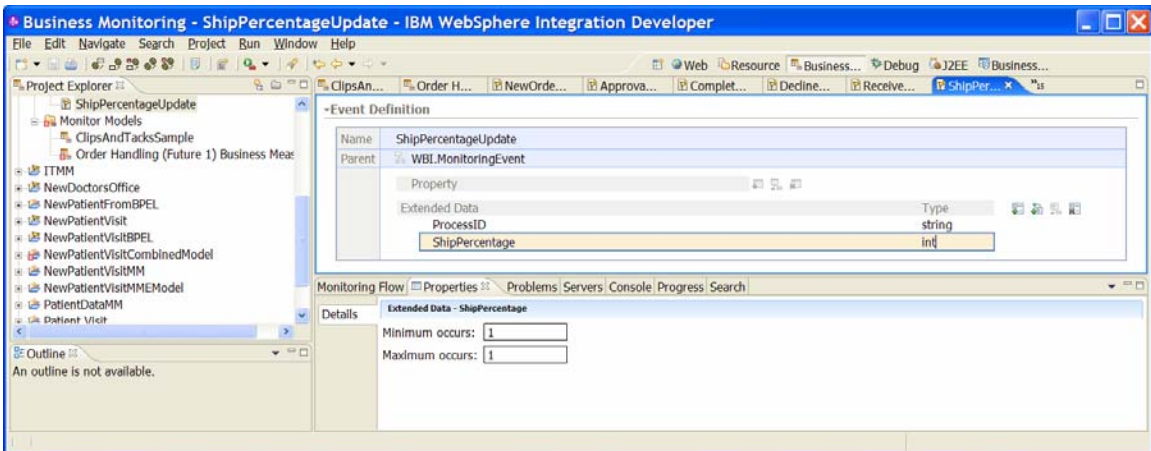
Definition of the **ReceiveOrder**:

Figure 28. Definition of the ReceiveOrder



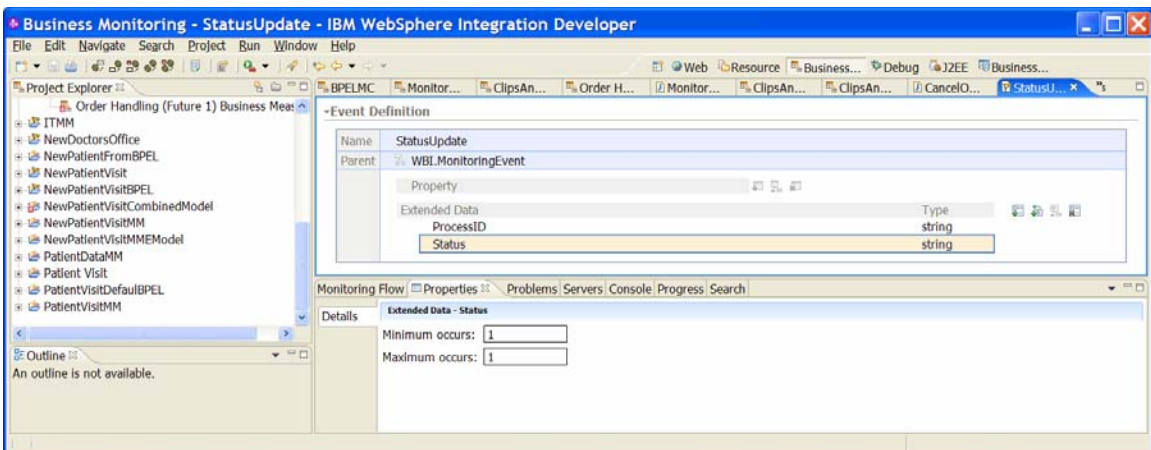
Definition of **ShipPercentageUpdate**:

Figure 29. Definition of ShipPercentageUpdate



Definition of **StatusUpdate**:

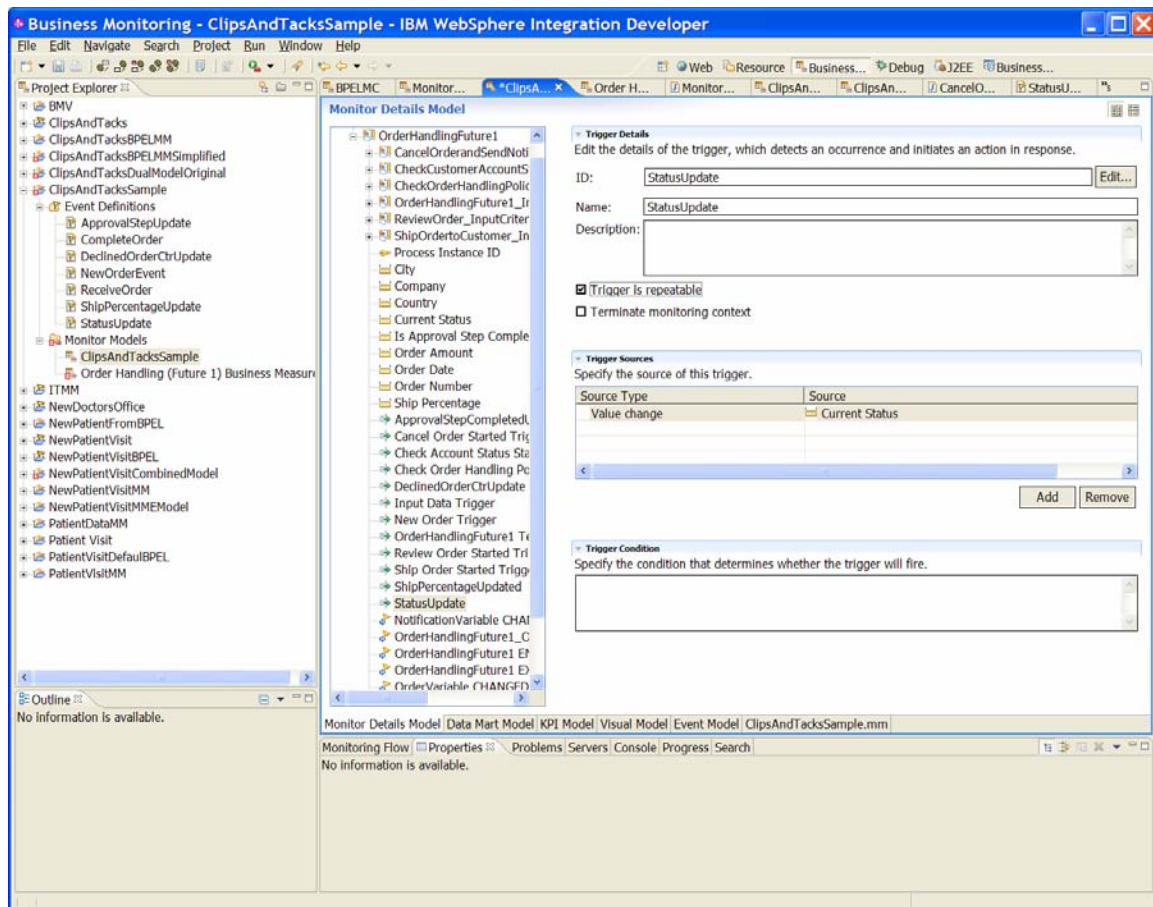
Figure 30. Definition of StatusUpdate



2. Now you need to add the triggers that cause these outbound events to be sent. Add these triggers:
- **ApprovalStepCompletedUpdate**: fires when the value for the Is Approval Step Completed metric changes
  - **DeclinedOrderCtrUpdated**: fires when the value for the Declined Order Counter counter changes
  - **ShipPercentageUpdated**: fires when the value for the Ship Percentage metric changes
  - **StatusUpdate**: fires when the value for the Current Status metric changes
  - **CountryUpdate**: fires when the value for the Country metric changes
  - **OrderAmountUpdate**: fires when the values for the Order Amount metric changes

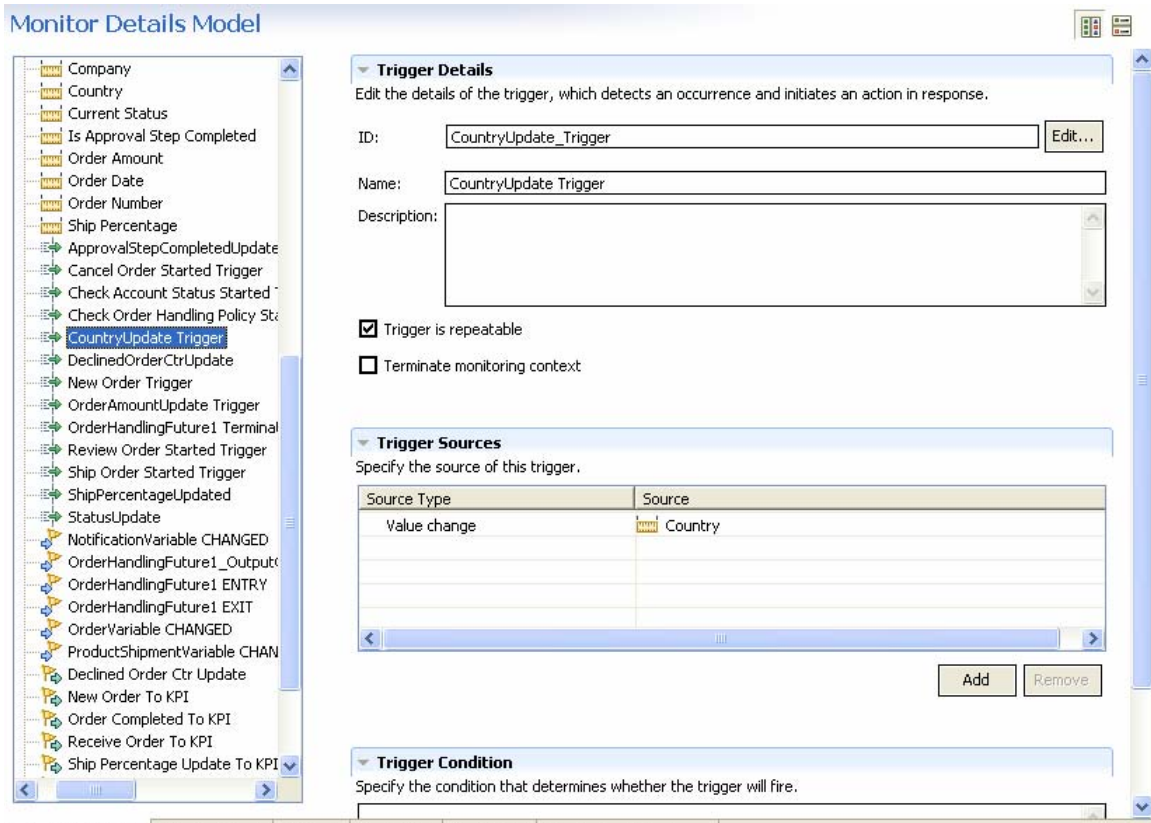
Here are examples of the **StatusUpdate**, **CountryUpdate** and **OrderAmountUpdate** triggers. Note that the Trigger is repeatable check box is selected, causing the trigger to fire every time the status changes. It is a good idea to select this option for all of the triggers mentioned previously. All of these triggers are added to the parent monitor context, OrderHandlingFuture1.

Figure 31. The StatusUpdate trigger



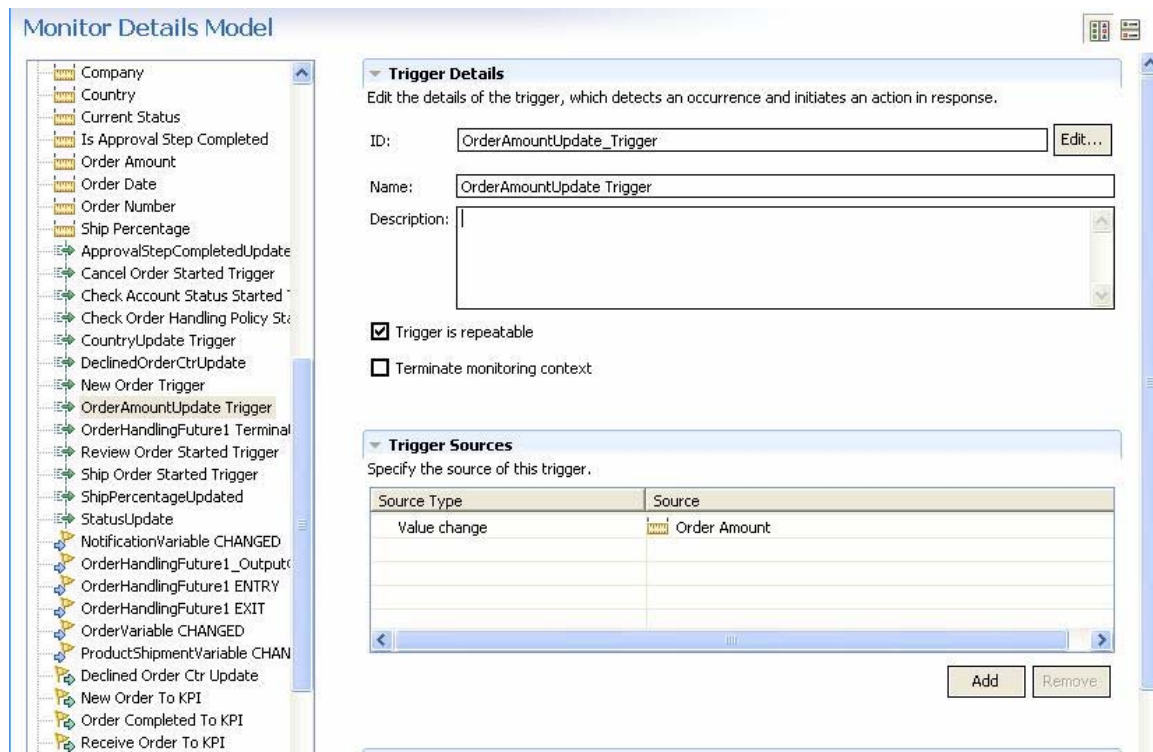
The CountryUpdate trigger:

Figure 32. The CountryUpdate\_Trigger



## The OrderAmountUpdate\_Trigger:

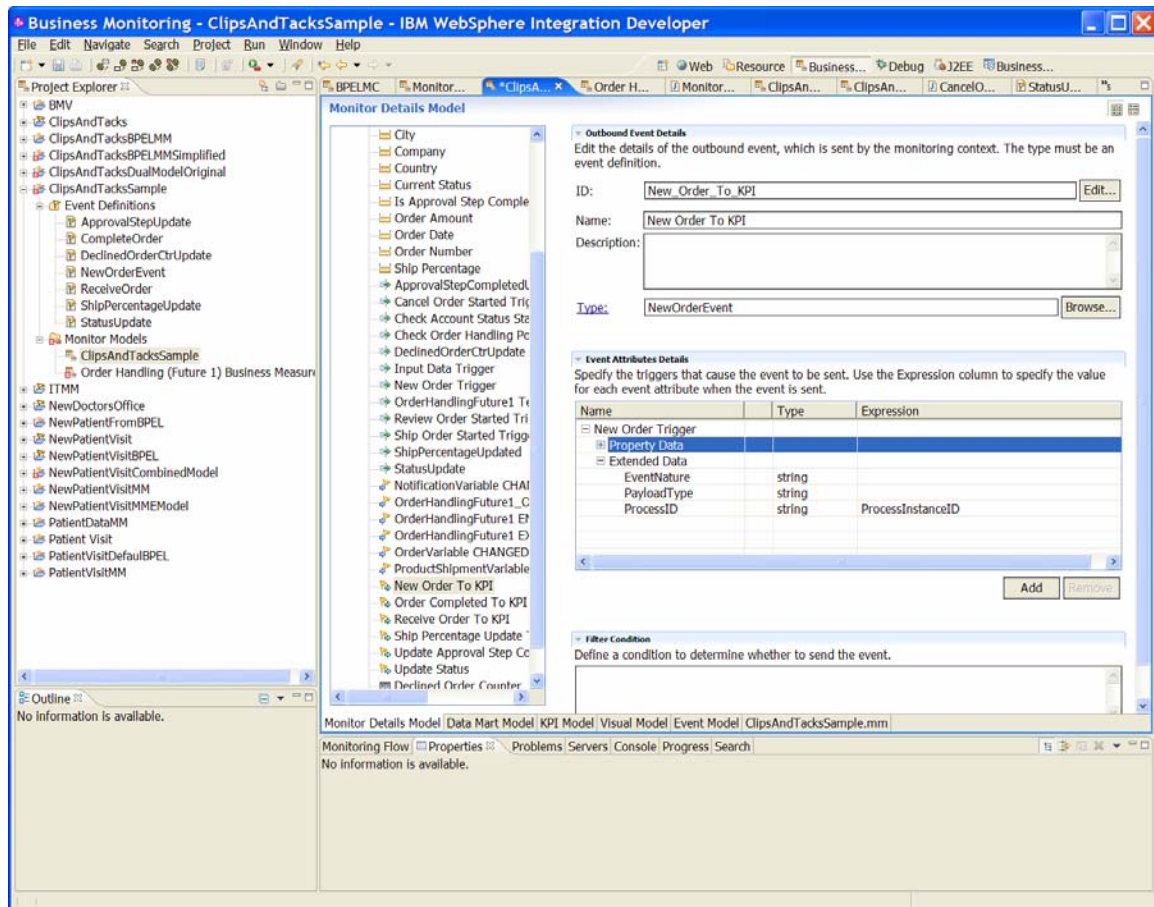
Figure 33. The OrderAmountUpdate\_Trigger



- \_\_\_\_\_ 3. Now that the triggers are created, you must create these outbound events:
- **New Order To KPI** (type: NewOrderEvent). Sent when the New Order Trigger fires.
  - **Order Completed To KPI** (type: CompleteOrder). Sent when the OrderHandlingFuture1 Termination Trigger fires.
  - **Receive Order To KPI** (type: ReceiveOrder). Sent when the Input Data Trigger fires.
  - **Ship Percentage Update To KPI**. This is the ShipPercentageUpdate event that is sent when the ShipPercentageUpdate trigger fires.
  - **Update Approval Step Completion**. This is the ApprovalStepUpdate event that is sent when the ApprovalStepCompletedUpdate fires.
  - **Update Status**. This is the StatusUpdate event that is sent when the New Order Trigger or the StatusUpdate fires.
  - **Declined Order Ctr Update**. This is the DeclinedOrderCtrUpdate event that is sent when the DeclinedOrderCtrUpdate fires.
  - **Update Country**. This is the CountryUpdate event that is sent when the New Order Trigger or CountryUpdate trigger fires.
  - **Update OrderAmount**. This is the OrderAmountUpdate event that is sent when the OrderAmountUpdate trigger fires.

The following figures show the definition for each of these outbound events:

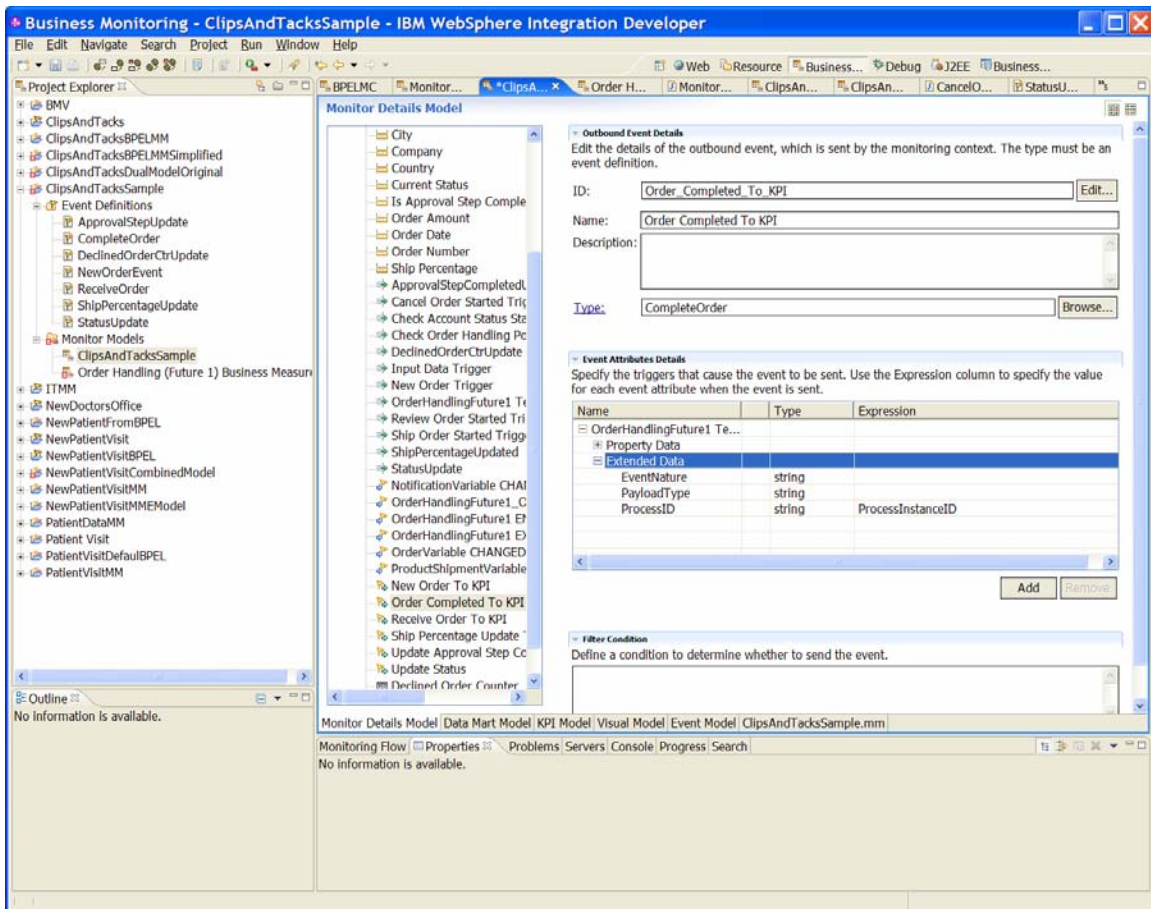
Figure 34. The New\_Order\_To\_KPI outbound event definition





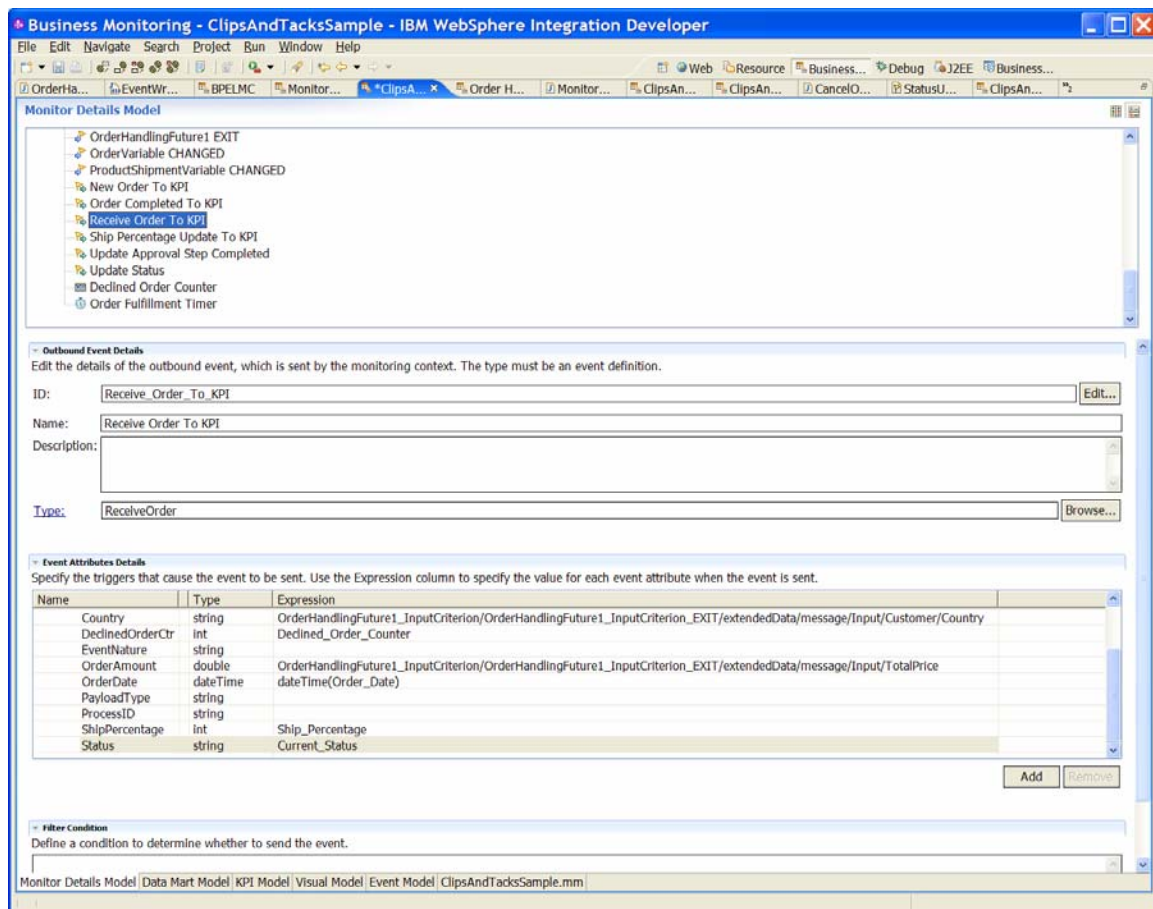
The Order\_Completed\_To\_KPI outbound event definition:

Figure 35. The Order\_Completed\_To\_KPI outbound event definition



## The Receive\_Order\_To\_KPI outbound event definition:

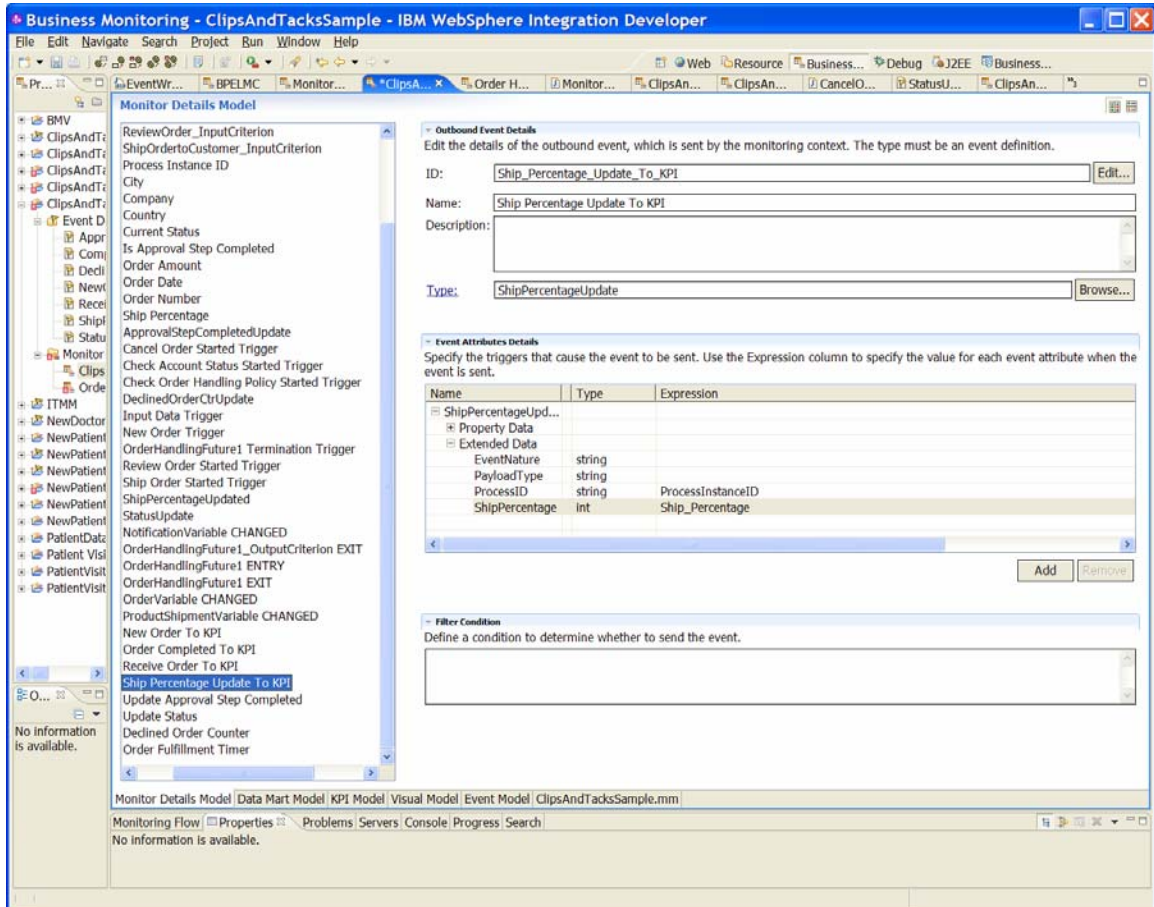
Figure 36. The Receive\_Order\_To\_KPI outbound event definition



For the preceding ReceiveOrderToKPI outbound event, notice that the Country and Order Amount attributes are set from the inbound event and not the corresponding metric. Because these metrics are set on the inbound event that is the source trigger for this outbound event, there is no guarantee that these values are set when the outbound event is sent.

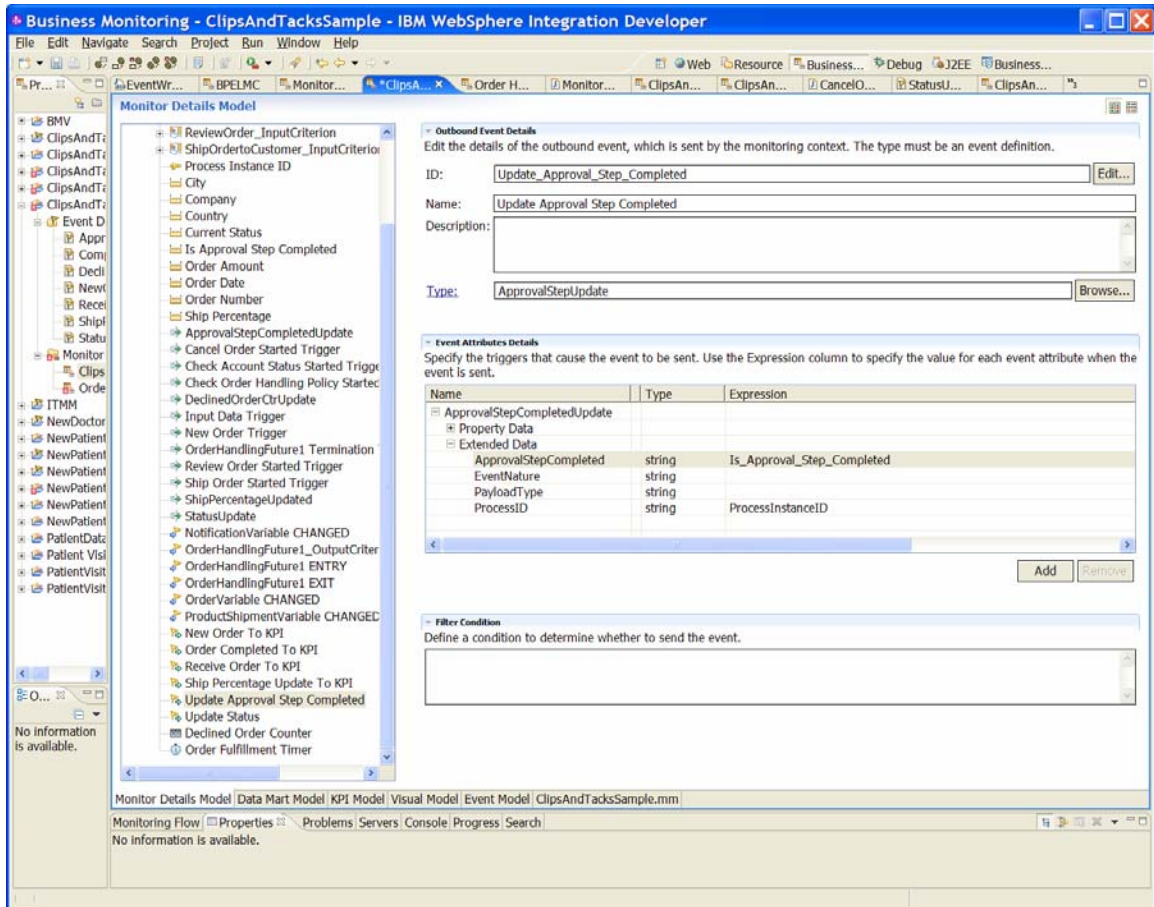
Therefore, you must use the values from the inbound event. The other values in the outbound event have been set by either default values or other inbound events.

Figure 37. Ship\_Percentage\_Update\_To\_KPI values set by defaults or inbound events



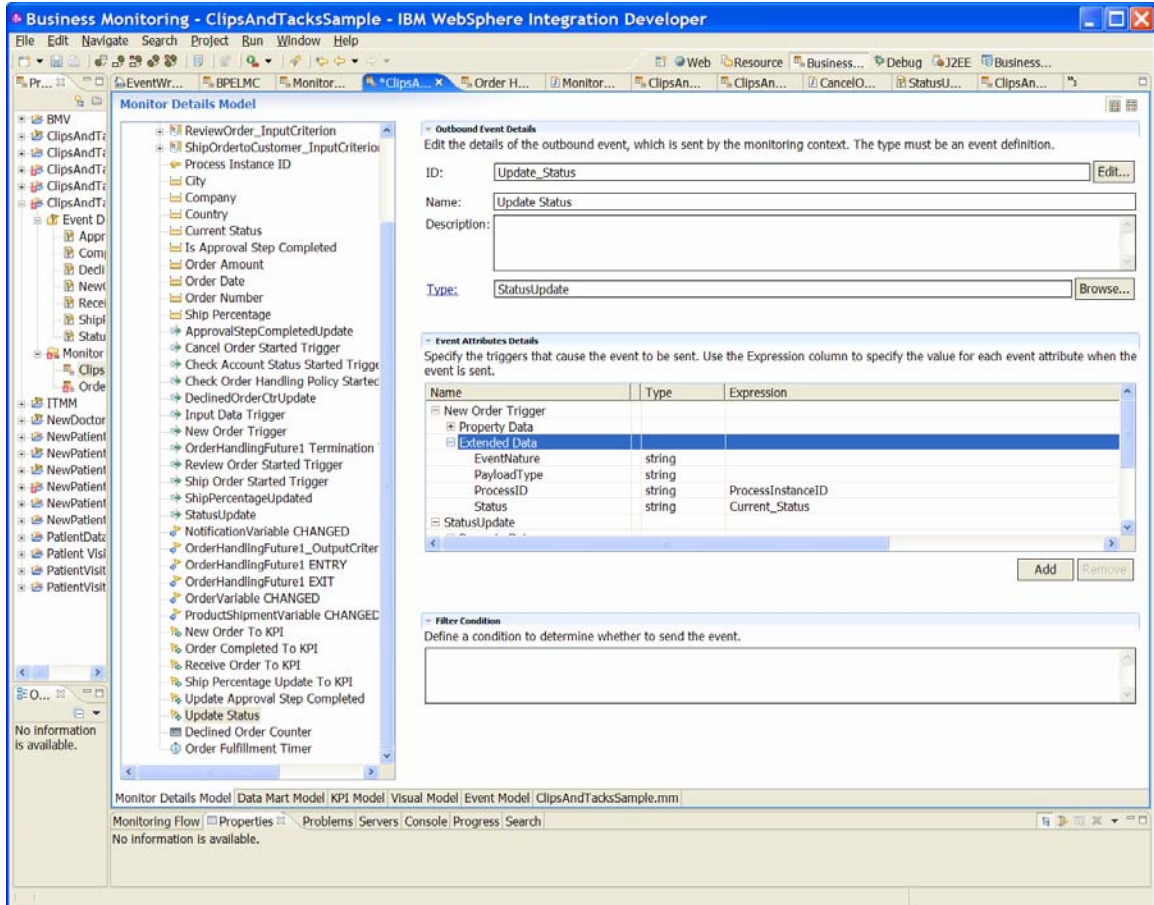
**Update\_Approval\_Step\_Completed** event values set by default values or inbound events:

Figure 38. Update\_Approval\_Step\_Completed values set by defaults or inbound events



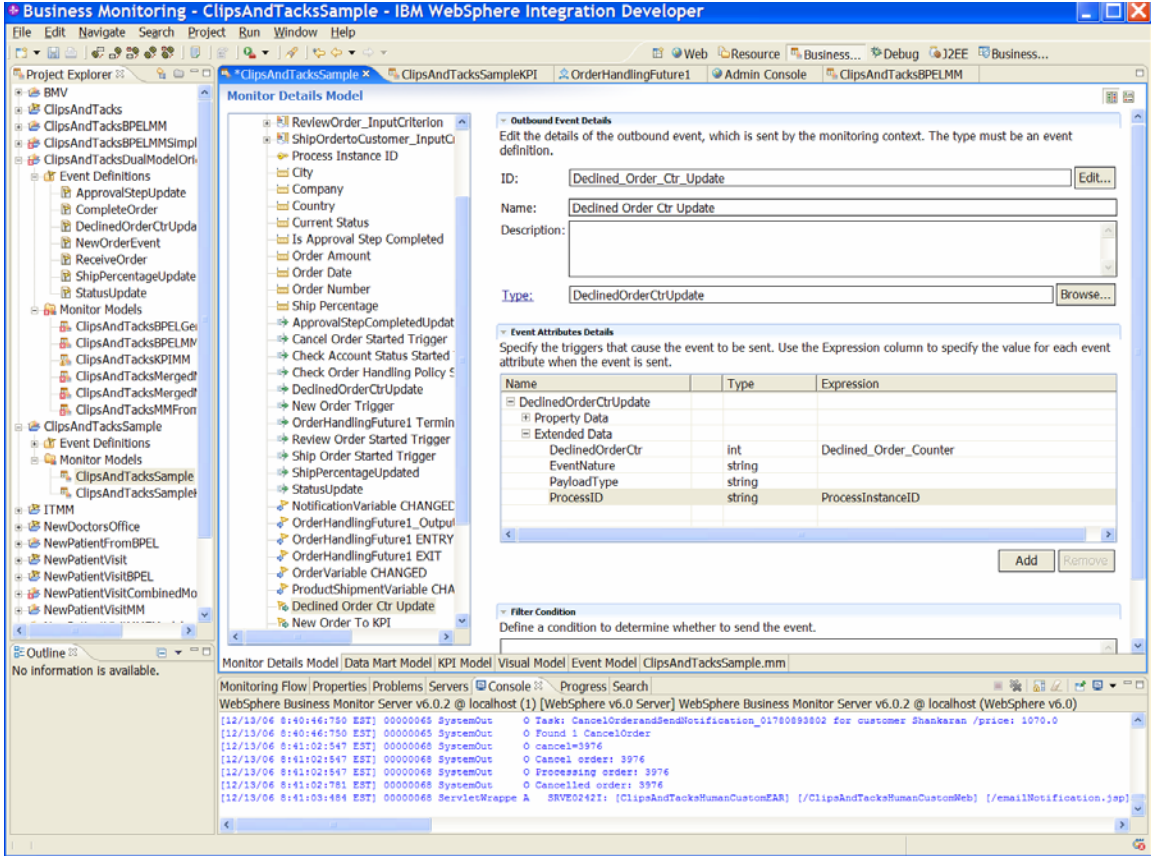
**Update\_Status event values** set by default values or inbound events:

Figure 39. Update\_Status event values set by default values or inbound events



There are two triggers for the preceding **UpdateStatus** outbound event.

Figure 40. Triggers that cause the Declined\_Order\_Ctr\_Update event



Triggers that cause the **Update\_Country** event:

Figure 41. Triggers that cause the Update\_Country event

The screenshot shows the 'Monitor Details Model' interface. On the left is a tree view of various triggers. The main area is titled 'GENERAL' and contains the following fields:

- ID:** Update\_Country
- Name:** Update\_Country
- Description:** (empty text area)
- Type:** CountryUpdate

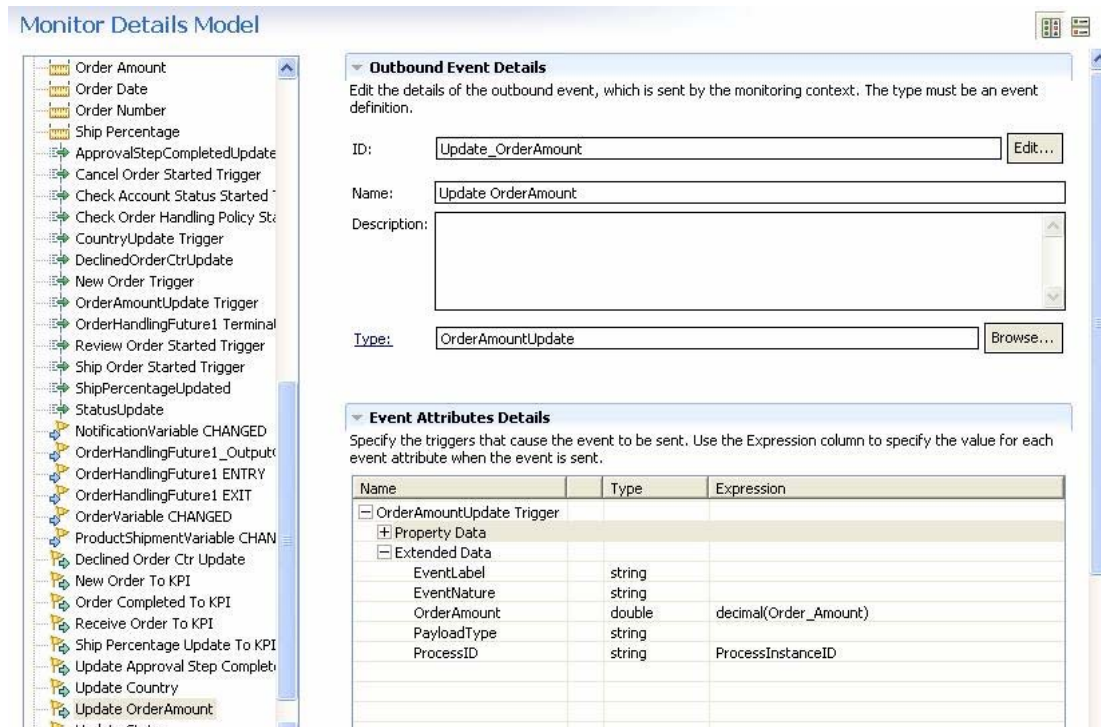
Below these fields is the 'Event Attributes Details' section, which includes a table for specifying triggers and their attributes:

Specify the triggers that cause the event to be sent. Use the Expression column to specify the value for each event attribute when the event is sent.

| Name                    | Type   | Expression        |
|-------------------------|--------|-------------------|
| + New Order Trigger     |        |                   |
| - CountryUpdate Trigger |        |                   |
| + Property Data         |        |                   |
| - Extended Data         |        |                   |
| Country                 | string | Country           |
| EventLabel              | string |                   |
| EventNature             | string |                   |
| PayloadType             | string |                   |
| ProcessID               | string | ProcessInstanceID |

Triggers that cause the **Update\_OrderAmount** event:

Figure 42. Triggers that cause the Update\_OrderAmount event

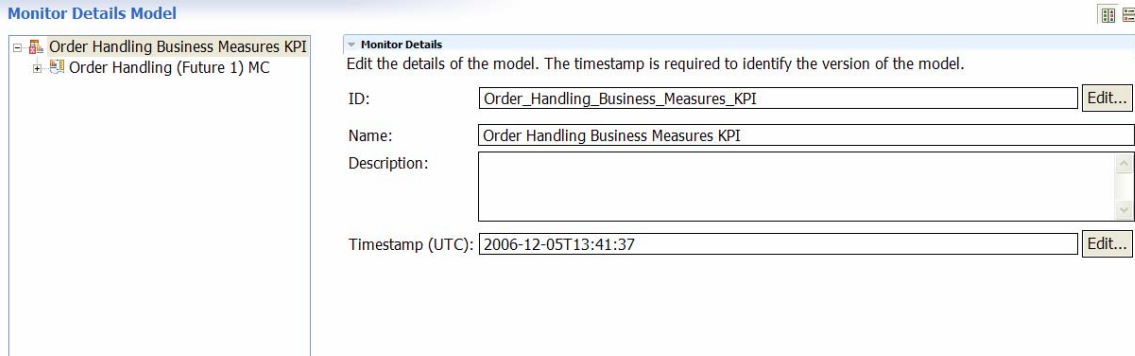


That completes the work in the BPEL model.



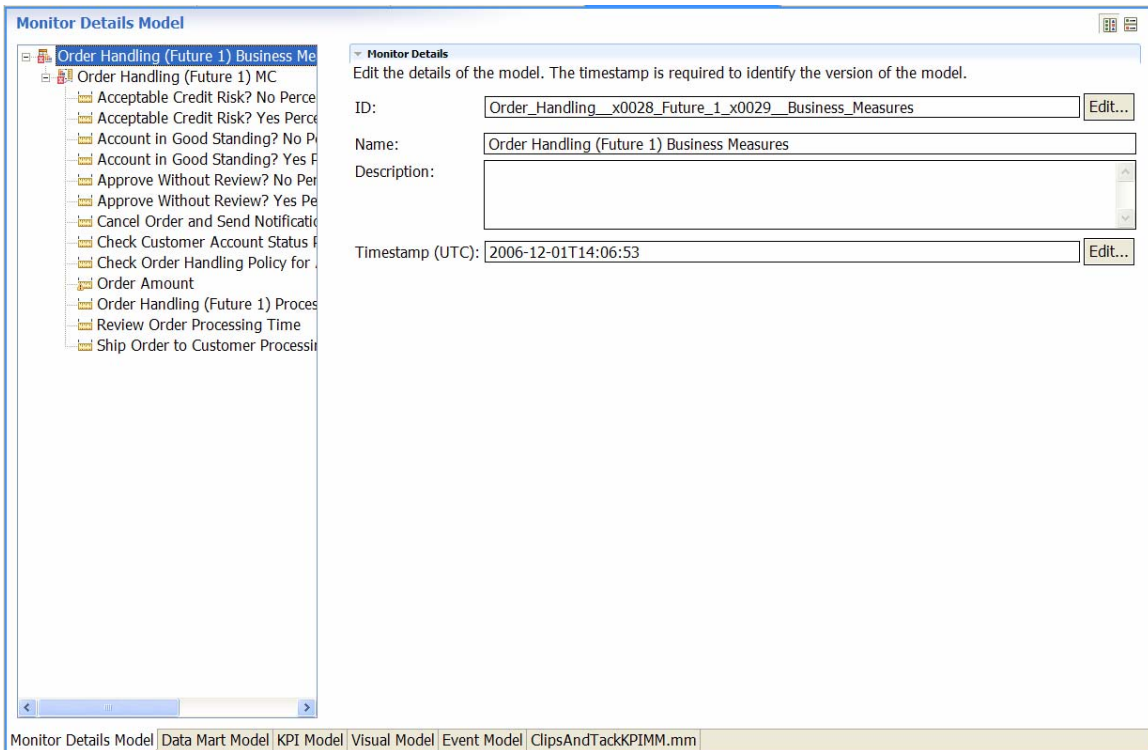
- \_\_\_\_\_ 4. Now, update the KPI model to handle the inbound events. For example, the generated model can provide Model Name and Model ID values with some cryptic characters, so change the ID and Name to make them easier to read:

Figure 43. Change the ID and Name to make them easier to read



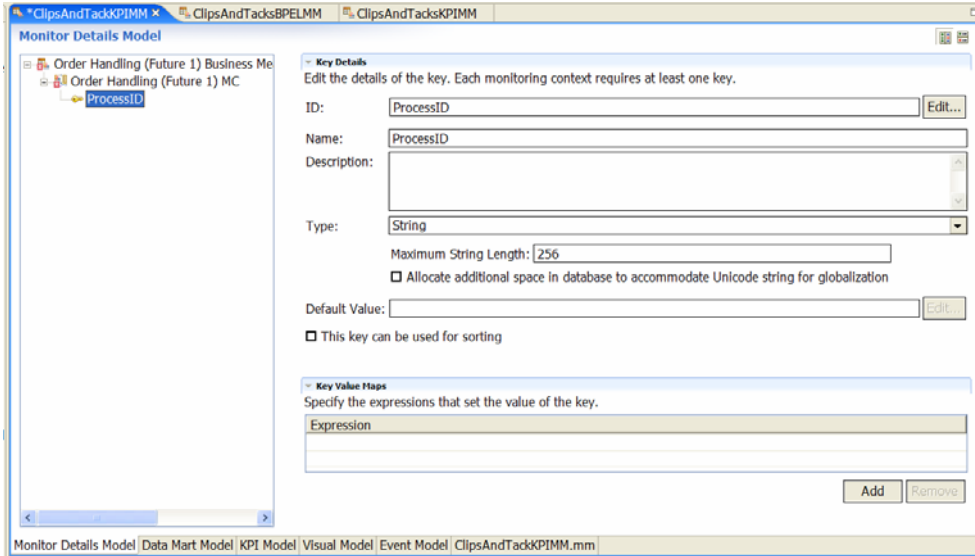
- \_\_\_\_\_ 5. Change the ID and Name of the **Order Handling (Future 1) MC** context to make it easier to read. Here is the model with the default metrics. First, remove all of the metrics that were under the parent context **Order Handling (Future 1) MC**.

Figure 44. Model with the default metrics that require removal



- \_\_\_\_\_ 6. Before you add the inbound events, add the key metric to the parent context. The key metric is used in the correlation expression of the events to make sure that the event is associated with the correct context. The key metric for the parent context is the Process ID value that is sent in the **NewOrderInboundEvent**. All inbound events sent by the BPEL model are filtered by the key value. The following example shows the definition of the key metric:

Figure 45. Definition of the key metric

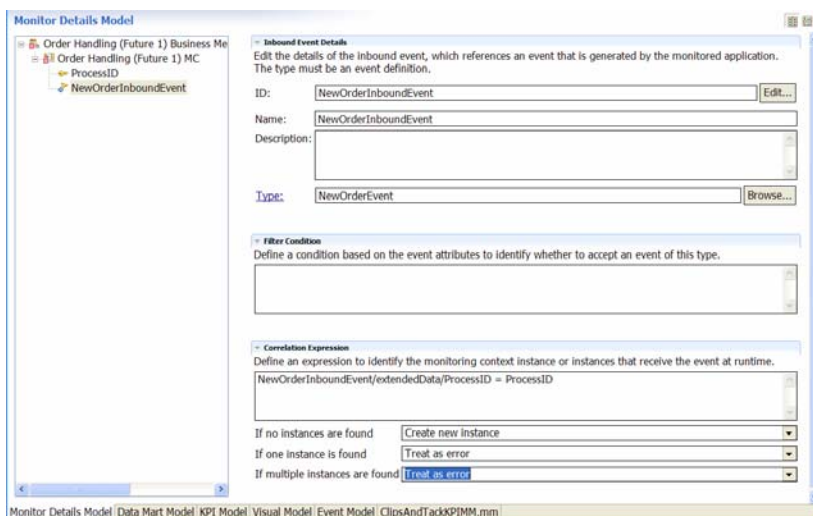


- \_\_\_\_\_ 7. Now add the following inbound events and the related correlation expressions:
- **NewOrderInboundEvent** (type: NewOrderEvent). The correlation expression verifies that the ProcessID in the event matches the ProcessID key metric in the context. If no instances are found for the correlation, a new instance should be created. If one or more instances are found, it should be treated as an error.
  - **Approval Update** (type: ApprovalStepUpdate). The correlation expression verifies that the ProcessID in the event matches the ProcessID key metric in the context. If one instance is found, the event should be delivered to the instance. If no instance or multiple instances are found, it should be treated as an error.
  - **End Order** (type: CompleteOrder). The correlation expression verifies that the ProcessID in the event matches the ProcessID key metric. If one instance is found, the event should be delivered to the instance. If no instance or multiple instances are found, it should be treated as an error.
  - **Receive Order** (type: ReceiveOrder). The correlation predicate expression verifies that the ProcessID in the event matches the ProcessID key metric in the context. If one instance is found, the event should be delivered to the instance. If no instance or multiple instances are found, it should be treated as an error.

- **Status Update** (type: StatusUpdate). The correlation expression verifies that the ProcessID in the event matches the ProcessID key metric in the context. If one instance is found, the event should be delivered to the instance. If no instance or multiple instances are found, it should be treated as an error.
- **Update Declined Order Counter** (type: DeclinedOrderCtrUpdate). The correlation expression verifies that the ProcessID in the event matches the ProcessID key metric in the context. If one instance is found, the event should be delivered to the instance. If no instance or multiple instances are found, it should be treated as an error.
- **Update Ship Percentage** – This inbound event is of type ShipPercentageUpdate. The correlation expression must verify that the ProcessID in the event matches the ProcessID key metric in the context. If one instance is found, the event should be delivered to the instance. If no instance or multiple instances are found, it should be treated as an error.
- **Country Update** – This inbound event is of type CountryUpdate. The correlation expression must verify that the ProcessID in the event matches the ProcessID key metric in the context. If one instance is found, the event should be delivered to the instance. If no instance or multiple instances are found, it should be treated as an error.
- **OrderAmount Update** – This inbound event is of type OrderAmountUpdate. The correlation expression must verify that the ProcessID in the event matches the ProcessID key metric in the context. If one instance is found, the event should be delivered to the instance. If no instance or multiple instances are found, it should be treated as an error.

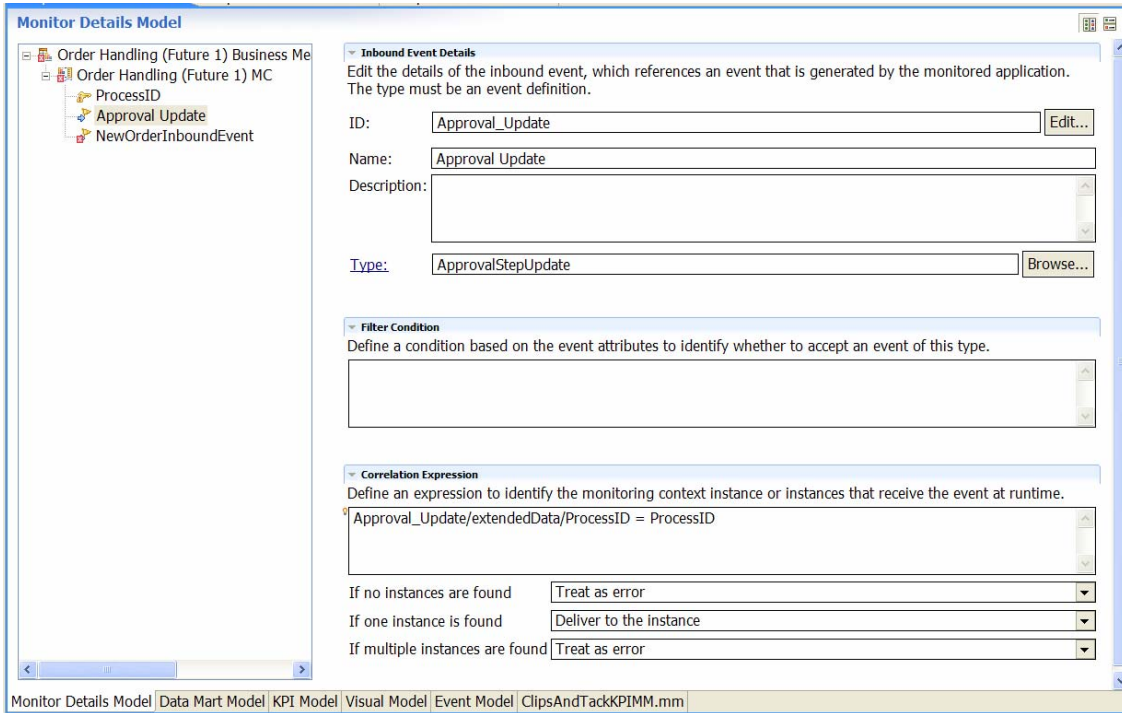
Here are the definitions for the **NewOrderInboundEvent** and the Approval Update inbound event showing the correlation expression and the actions for number of instances found:

Figure 46. Correlation expression and actions for no. of NewOrderInboundEvent instances



Approval Update inbound event definition showing the correlation expression and the actions for number of instances found:

Figure 47. Correlation expression and the actions for number of instances for Approval Update

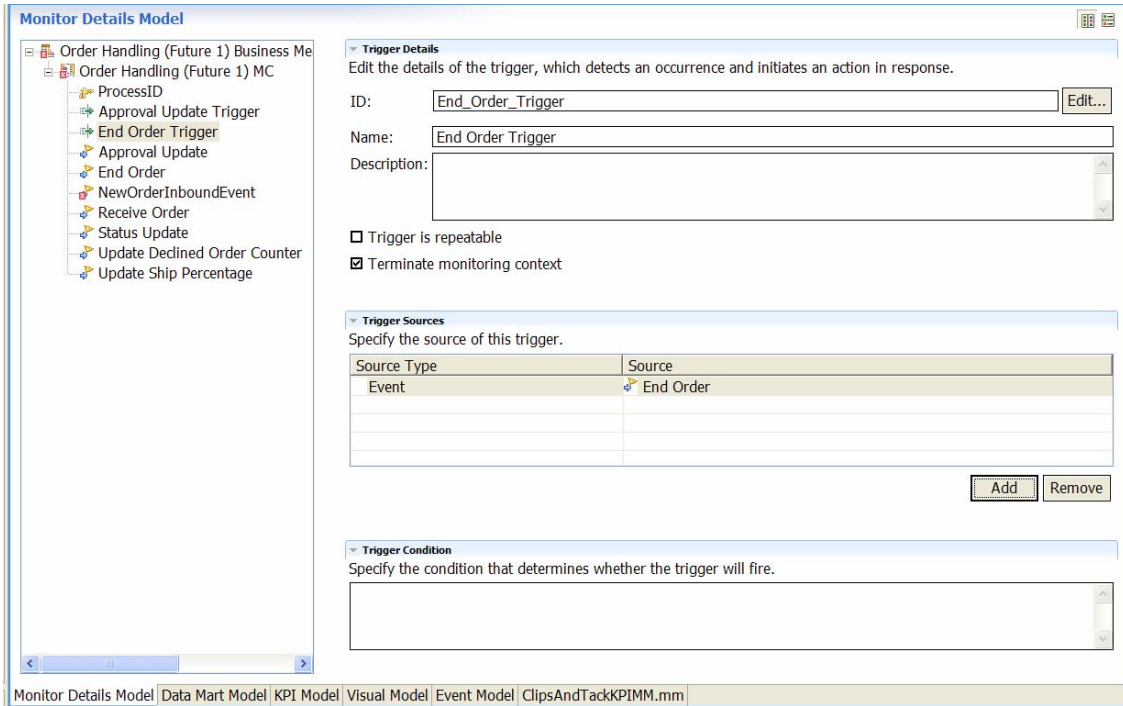


8. Now that you have created the events, you can create the following triggers for each of the events:

- **Approval Update Trigger** – Source Type for this trigger is the Approval Update inbound event.
- **End Order Trigger** – Source Type for this trigger is the End Order inbound event. This trigger also terminates the monitor context.
- **New Order Trigger** – Source Type for this trigger is the NewOrderInboundEvent.
- **Receive Order Trigger** – Source Type for this trigger is the Receive Order inbound event.
- **Status Update Trigger** – Source Type for this trigger is the Status Update inbound event. Because multiple status updates are received as the order is processed, select the **Trigger is repeatable** check box for this trigger.
- **Update Ship Percentage Trigger** – Source Type for this trigger is the Update Ship Percentage inbound event.
- **Declined Order Counter Trigger** – Source Type for this trigger is the Update Declined Order Counter inbound event.
- **Country Update Trigger** – Source Type for this trigger is the Country Update inbound event.
- **OrderAmount Update Trigger** – Source Type for this trigger is the OrderAmount Update inbound event.

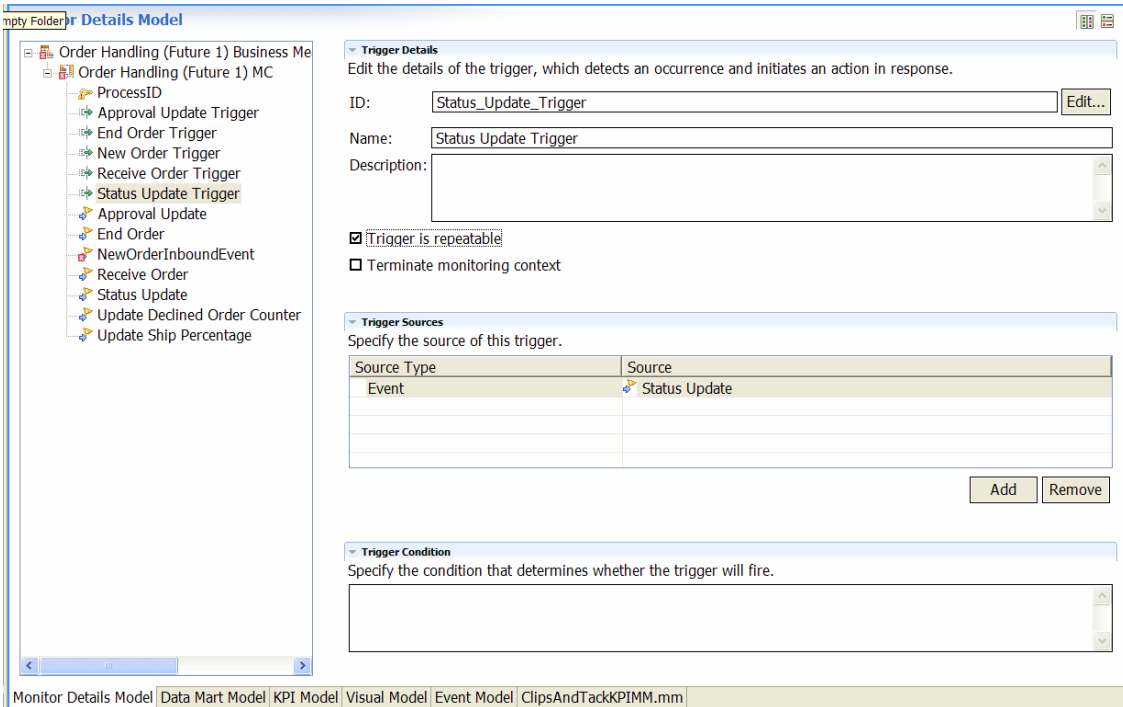
Here are the definitions for the **End Order Trigger** and the **Status Update Trigger**:

Figure 48. End Order Trigger definitions



Definitions for the **Status Update Trigger**:

Figure 49. Definitions for Status Update Trigger



Definition for the **Country Update Trigger**:

Figure 50. Definition for Country Update trigger

The screenshot shows the 'Monitor Details Model' configuration window. On the left is a tree view of the 'Order Handling Business Measures' model, with 'Country Update Trigger' selected. The main area contains the following configuration fields:

- Name:** Country Update Trigger
- Description:** (Empty text box)
- Trigger is repeatable
- Terminate monitoring context

The **Trigger Sources** section is expanded, showing a table with the following data:

| Source Type | Source         |
|-------------|----------------|
| Event       | Country Update |
|             |                |
|             |                |
|             |                |
|             |                |

Below the table are 'Add' and 'Remove' buttons. The **Trigger Condition** section is also expanded, with a text box for specifying the condition that determines whether the trigger will fire.

Definition for the **OrderAmount Update Trigger**:

Figure 51. Definition for OrderAmount Update trigger

The screenshot shows the 'Monitor Details Model' configuration window. On the left is a tree view of the model structure. The main area is divided into sections for defining the trigger.

**Name:** OrderAmount Update Trigger

**Description:** [Empty text box]

Trigger is repeatable

Terminate monitoring context

**Trigger Sources**  
Specify the source of this trigger.

| Source Type | Source             |
|-------------|--------------------|
| Event       | OrderAmount Update |
|             |                    |
|             |                    |
|             |                    |
|             |                    |

[Add] [Remove]

**Trigger Condition**  
Specify the condition that determines whether the trigger will fire.

[Empty text box]

- \_\_\_\_ 9. Now that you have added the triggers, add these metrics, counters, and stopwatches for the KPIs:
- **Country:** metric, string of length 256, default value of US; value is set from the Country field in the Receive Order event or the Country Update event when the when the Receive Order Trigger or Country Update trigger is fired. (Add the two triggers to Metric Value Maps.)
  - **Current Status:** metric, string of length 256, no default value; value is set from the Status field in the Status Update event or the Receive Order event when the Status Update Trigger or Receive Order Trigger is fired. The value is set to Started when the New Order Trigger is fired. (Add the three triggers to Metric Value Maps.)
  - **Is Approval Step Completed:** metric, length of 256, default value of Not Completed; value is set from the ApprovalStepCompleted field in the Approval\_Update event when the Approval Update Trigger is fired, and from the ApprovalStepCompleted field in the Receive\_Order event when the Receive Order Trigger is fired.

- **Order Amount:** decimal, default value of 0; value is set from the OrderAmount field in the Receive\_Order event or OrderAmount Update event when the Receive Order Trigger or OrderAmount Update trigger is fired. (Add the two triggers to Metric Value Maps.)
- **Order Date:** date, no default value; value is set from the OrderDate field in the Receive Order event when the Receive Order Trigger is fired. It is also set from the creationTime field in the NewOrderInboundEvent event. (Add the two triggers to Metric Value Maps.) Because the OrderDate field in the event has a type of DateTime, the expression must use the data() function to convert this to a date field.
- **Ship Percentage:** integer, defaults to 0; value is set from the ShipPercentage field in the Receive\_Order event or the Update\_Ship\_Percentage event when the Receive Order Trigger or Update Ship Percentage Trigger fires.
- **Declined Order Counter:** integer, defaults to 0; value is set from the DeclinedOrderCtr field in the DeclinedOrderCtrUpdate event when the Declined Order Counter Update Trigger is fired.
- **Order Fulfillment Timer:** stopwatch; started when the NewOrderEvent inbound event is received and stopped when the End Order inbound event is received.



Here are the definitions for the **Order Date** and the **Order Fulfillment Timer**:

Figure 52. Order Date Timer definition

The screenshot shows the 'Monitor Details Model' window. On the left is a tree view of the model structure. The 'Order Date' metric is selected. The main area shows the 'Metric Details' section with the following fields:

- ID: Order\_Date
- Name: Order Date
- Description: (empty)
- Type: Date
- Default Value: (empty)
- This metric can be used for sorting

The 'Metric Value Maps' section contains a table with one entry:

| Trigger               | Expression                                 |
|-----------------------|--|
| Receive Order Trigger | date(Receive_Order/extendedData/OrderDate) |

Buttons for 'Add' and 'Remove' are visible at the bottom right of the table.

The **Order Fulfillment Timer** definition:

Figure 53. Order Fulfillment Timer definition

The screenshot shows the 'Monitor Details Model' window. On the left is a tree view of the model structure. The 'Order Fulfillment Timer' stopwatch is selected. The main area shows the 'Stopwatch Details' section with the following fields:

- ID: Order\_Fulfillment\_Timer
- Name: Order Fulfillment Timer
- Description: (empty)
- This stopwatch is an accumulating stopwatch
- This stopwatch can be used for sorting

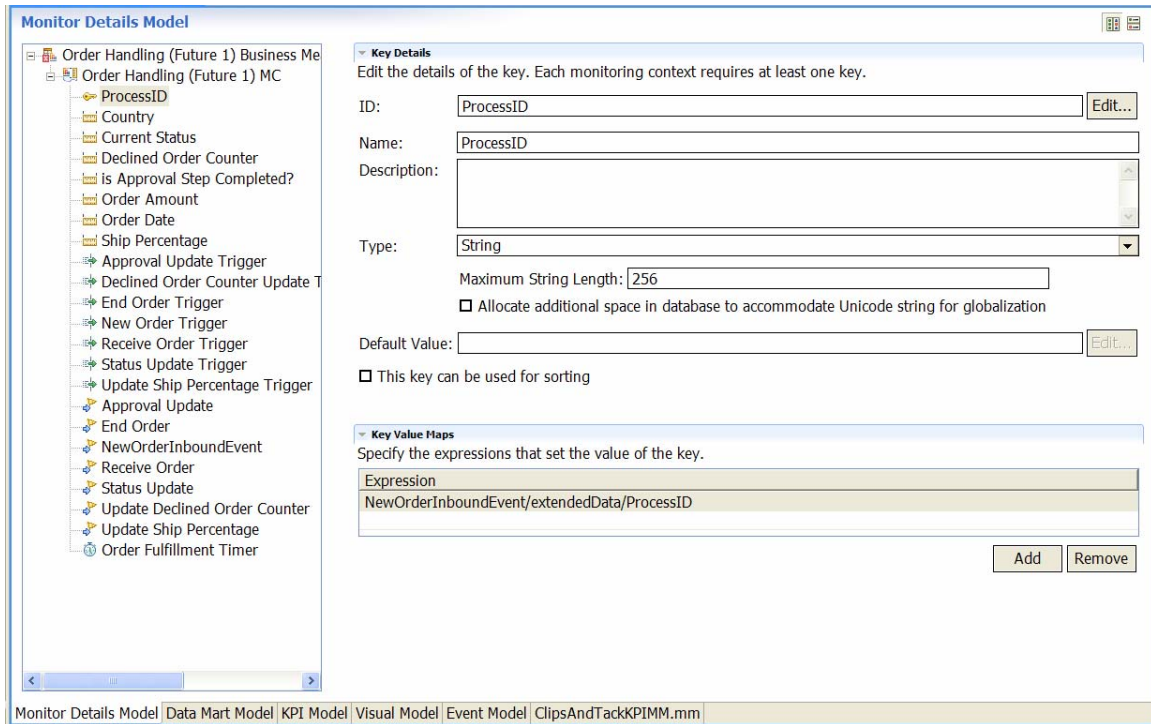
The 'Stopwatch Controls' section contains a table with two entries:

| Trigger / Inbound Event | Resulting Action |
|-------------------------|------------------|
| New Order Trigger       | Start            |
| End Order Trigger       | Stop             |

Buttons for 'Add' and 'Remove' are visible at the bottom right of the table.

10. You still need to set the value of the **ProcessID** key metric field. This field is from the **ProcessID** field in the **NewOrderInboundEvent** event. Update the Key Value Map for this key metric with the expression that sets the value of this key, as shown in the following example:

Figure 54. Set the value of the ProcessID key metric field



### Setting up the Data Mart model

Now you can work on the Data Mart model to set up the dimensions, facts, and measures.

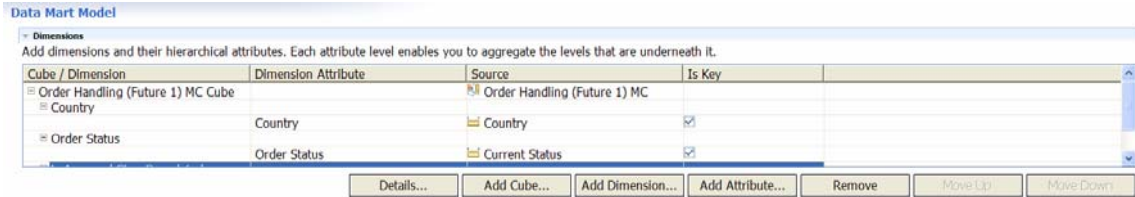
First, make sure that the following dimensions are defined on the **Order Handling** cube. The dimensions should already be specified (except for the **Order Date**), but the dimension attributes and dimension sources are missing. Because these dimensions are used in the KPI model, you need to complete the definitions in the Data Mart model. For **Order Date**, add the dimension as well as the attribute and source.

- **Country**: for Dimension Attribute and Source, specify Country; select the Is Key check box.
- **Order Status**: for Dimension Attribute, specify Order Status; for Source, specify Current Status; select the Is Key check box.
- **Order Date**: for Dimension Attribute and Source, specify Order Date; select the Is Key check box.

- **Is Approval Step Completed:** for Dimension Attribute and Source, specify Is Approval Step Completed; select the **Is Key** check box.

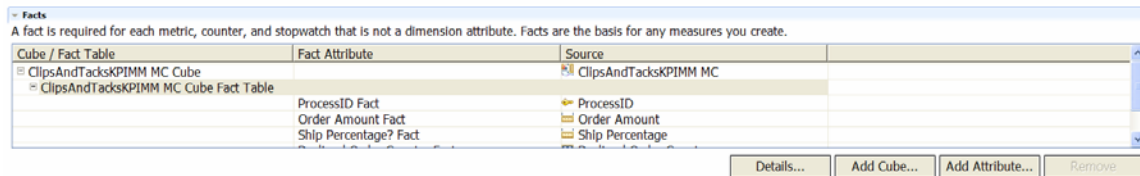
Here is the Data Mart model page showing the details for the Country and Order Status dimensions:

Figure 55. Data Mart model page details for Country and Order Status dimensions



Now you need to correct the facts for which a red x is displayed in the **Source** field. That leaves facts for **Process ID, Order Amount, Ship Percentage, Declined Order Counter, and Order Fulfillment Timer**. If these facts do not exist, add the facts using the metrics as shown in the figure. For example, for the fact **Order Amount**, the source is **Order Amount** metric. If the **Order Date** is still a fact, delete that as well. Some of these are shown here:

Figure 56. Correct facts for which x is displayed in the Source field

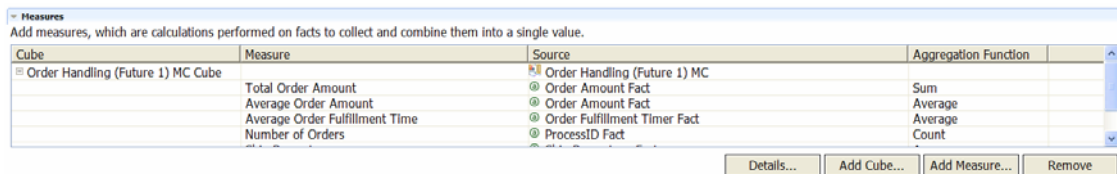


Now you need to ensure the measures are correct. After you renamed the model, the measures might contain some errors, which need to be fixed. Confirm that the following measures are defined with the indicated values:

- **Total Order Amount** – Source is Order Amount Fact and the Aggregation Function is Sum.
- **Average Order Amount** – Source is Order Amount Fact and the Aggregation Function is Average.
- **Average Order Fulfillment Time** – Source is Order Fulfillment Timer Fact and the Aggregation Function is Average
- **Number of Orders** – Source is ProcessID Fact and the Aggregation Function is Count.
- **Ship Percentage** – Source is Ship Percentage Fact and the Aggregation Function is Average.
- **Declined Order** – Source is Declined Order Counter Fact and the Aggregation Function is Sum.

Here is an example of the measures:

Figure 57. Example of the corrected measures



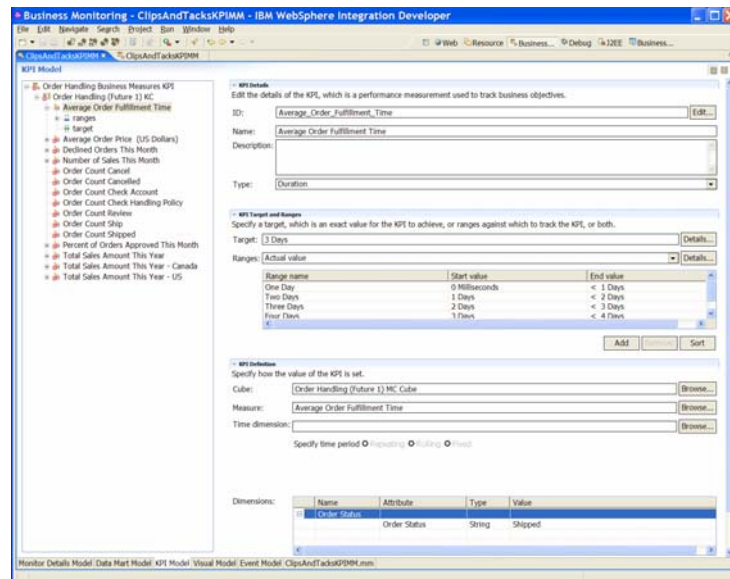
### Setting up the KPI model

1. Next, go to the KPI model view and complete the KPI model.

For the Average Order Fulfillment Time KPI, specify these values, as shown in the example:

- **Measure:** Average Order Fulfillment Time
- **Dimensions:** Order Status; add the Shipped value

Figure 58. Specify Measure and Dimension values

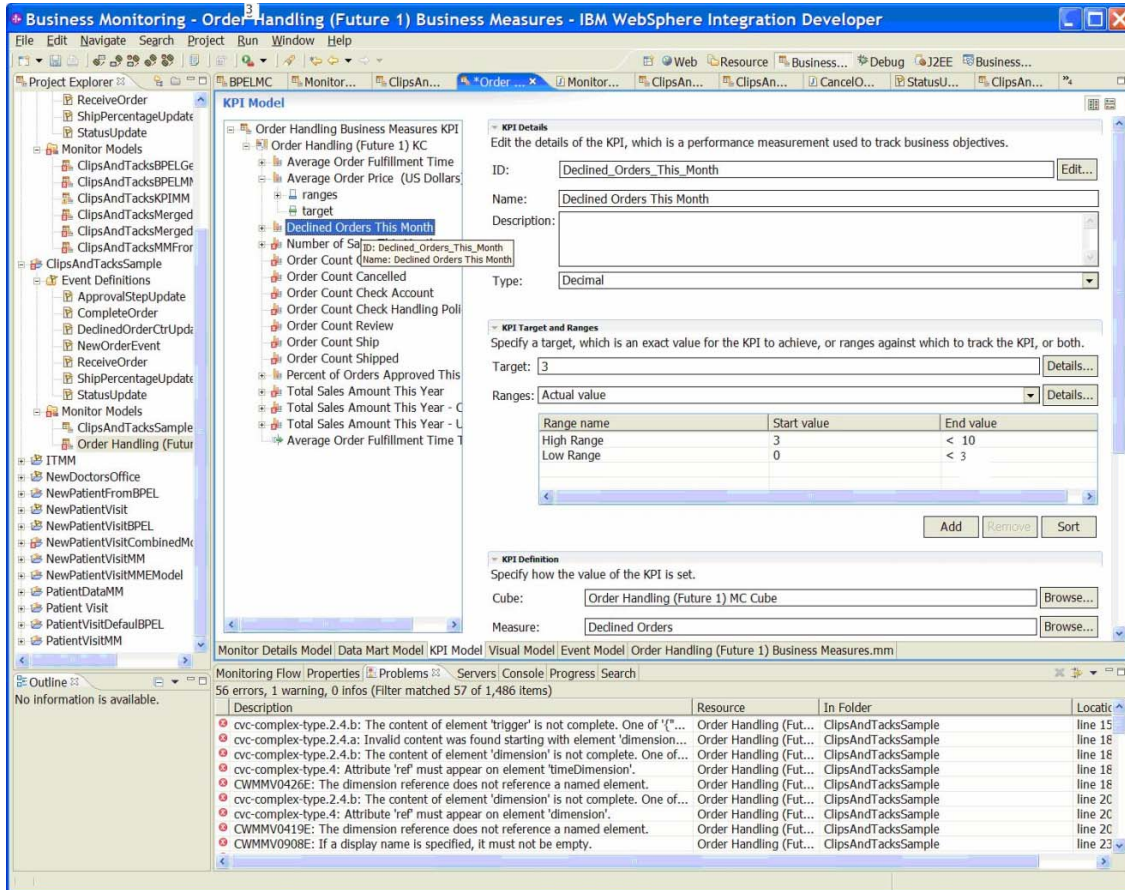


For the Average Order Price KPI, specify a measure of Average Order Amount. You need to change the **ID** and **name** to remove any cryptic characters. For example, change the ID to Average\_Order\_Price\_US\_Dollars. You might need to use the XML editor to delete an empty dimension tag that is flagged with an error.

For the Declined Orders This Month KPI, specify the **Declined Order Measure** and the **Order Date Time** dimension. Also make sure to specify the type of Decimal.

2. Verify that the **High Range** and **Low Range** values are specified; if not, specify the correct target and ranges for the KPI, as in this example:

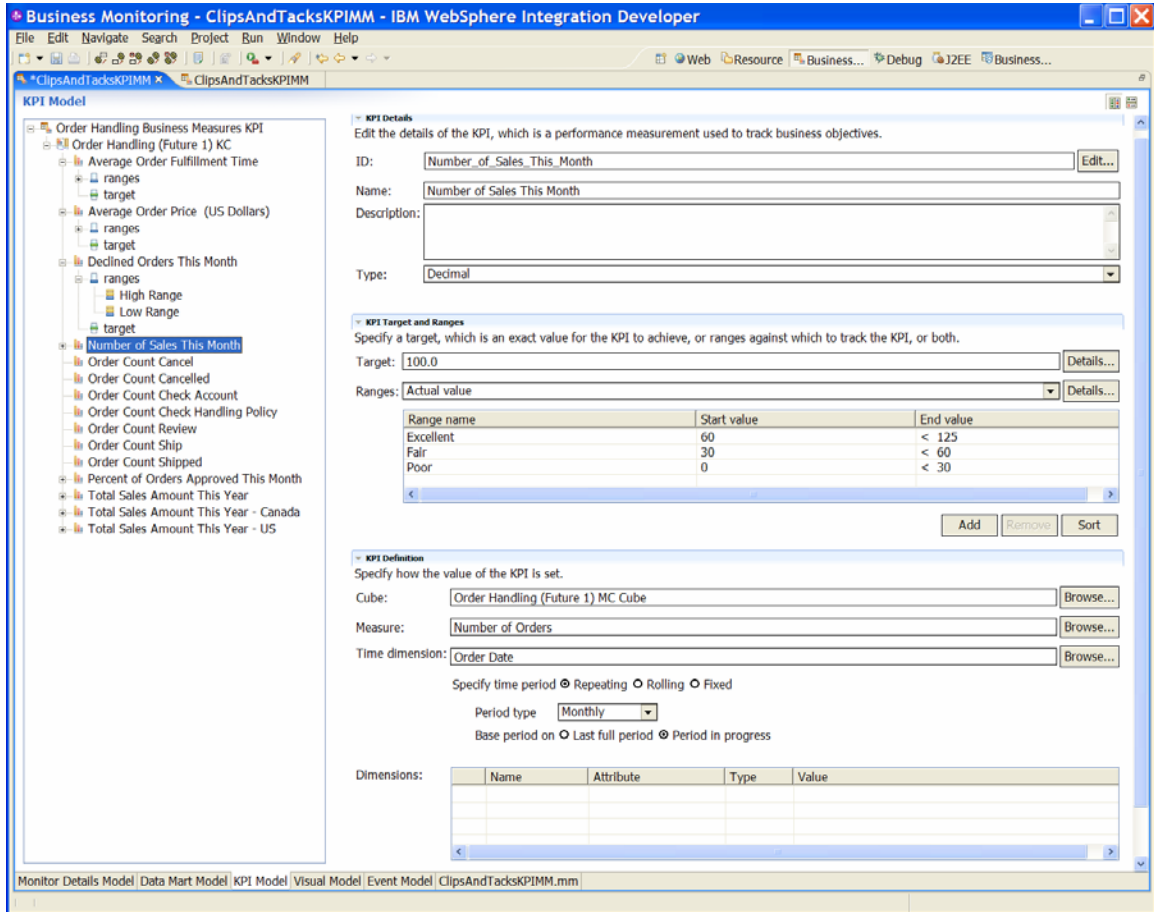
Figure 59. Specify correct target and ranges for the KPI



Based on the problems flagged in the preceding example, you might need to use the XML editor to delete the **displayName=""** attribute that is flagged with an error. After deleting the **displayName=""** attribute, rebuild the monitor project to avoid appearance of the error indicator in the xml editor view.

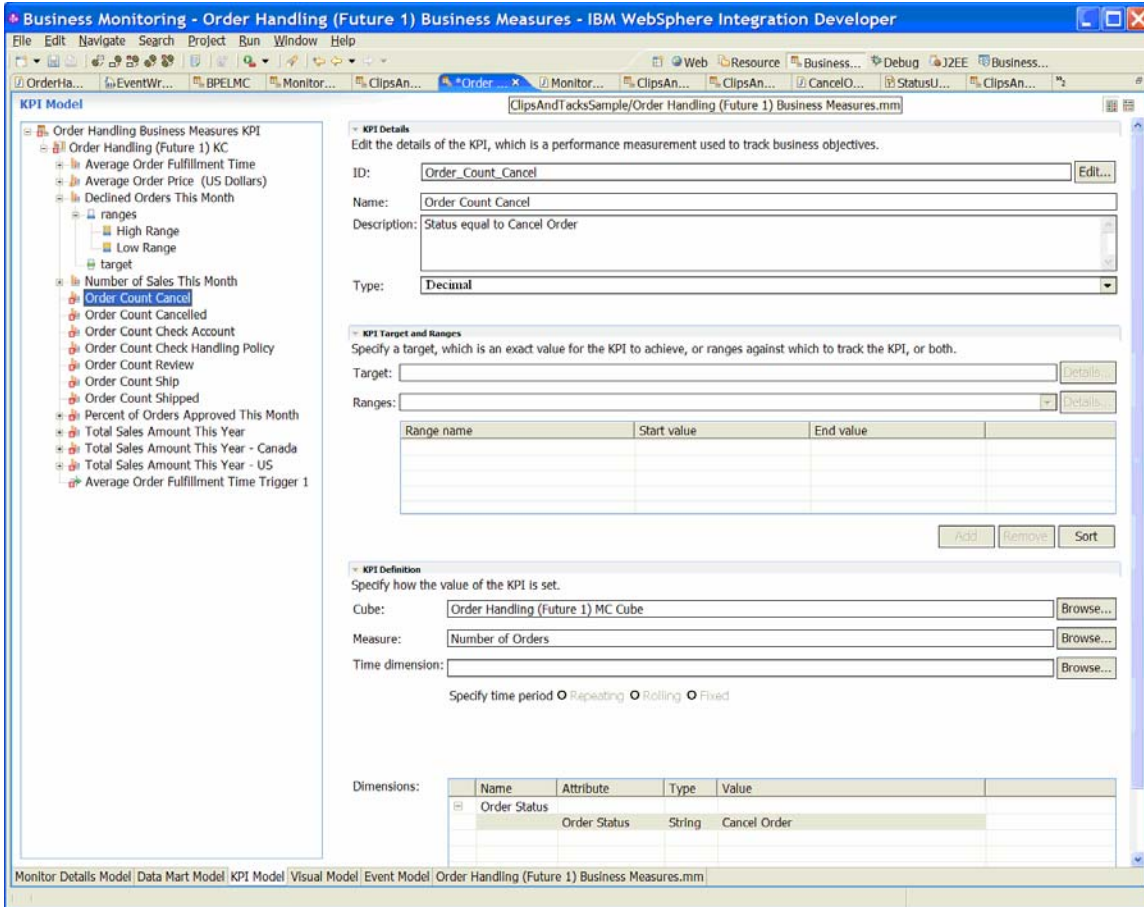
- 3. For the Number of Sales This Month KPI, add the **Number of Orders Measure** and the **Order Date Time** dimension as shown here:

Figure 60. Add Number of Orders Measure and Order Date Time dimension



- \_\_\_\_\_ 4. For the Order Count Cancel KPI, add the **Cancel Order** value string (do not place the value in quotation marks) to the **Order Status** dimension and make sure that **Decimal** is specified as the **Type**, as shown here:

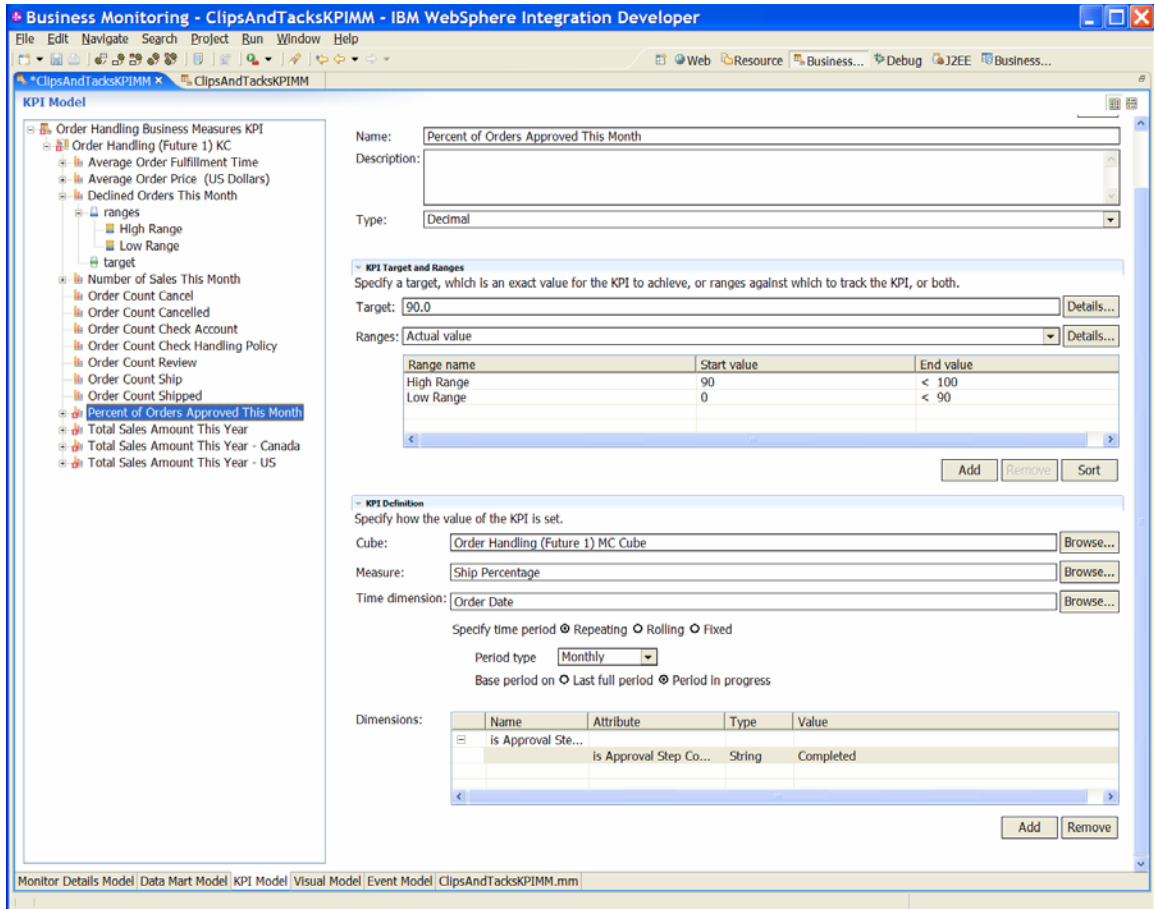
Figure 61. Assure Decimal is specified as the Type



- \_\_\_\_\_ 5. Now, make sure each of the following KPIs has a decimal type, has the Measure set to the Number of Orders, and this value in the Order Status dimension check:
- Order Count Cancelled - Cancelled
  - Order Count Check Account – Check Account
  - Order Count Check Handling Policy – Check Order Handling Policy
  - Order Count Review – Review Order
  - Order Count Ship – Ship Order
  - Order Count Shipped – Shipped

- \_\_\_\_\_ 6. For the Percent of Order Approved This Month KPI, change the **ID** and **Name** to be as short as possible. If the **Name** or **ID** is too long, it could cause problems during configuration of the KPI portlet. Then add the Ship Percentage as the **Measure**, add the **Order Date** as the **Time Dimension**, and add the Completed string to the **is Approval Step Completed** dimension. This is shown here:

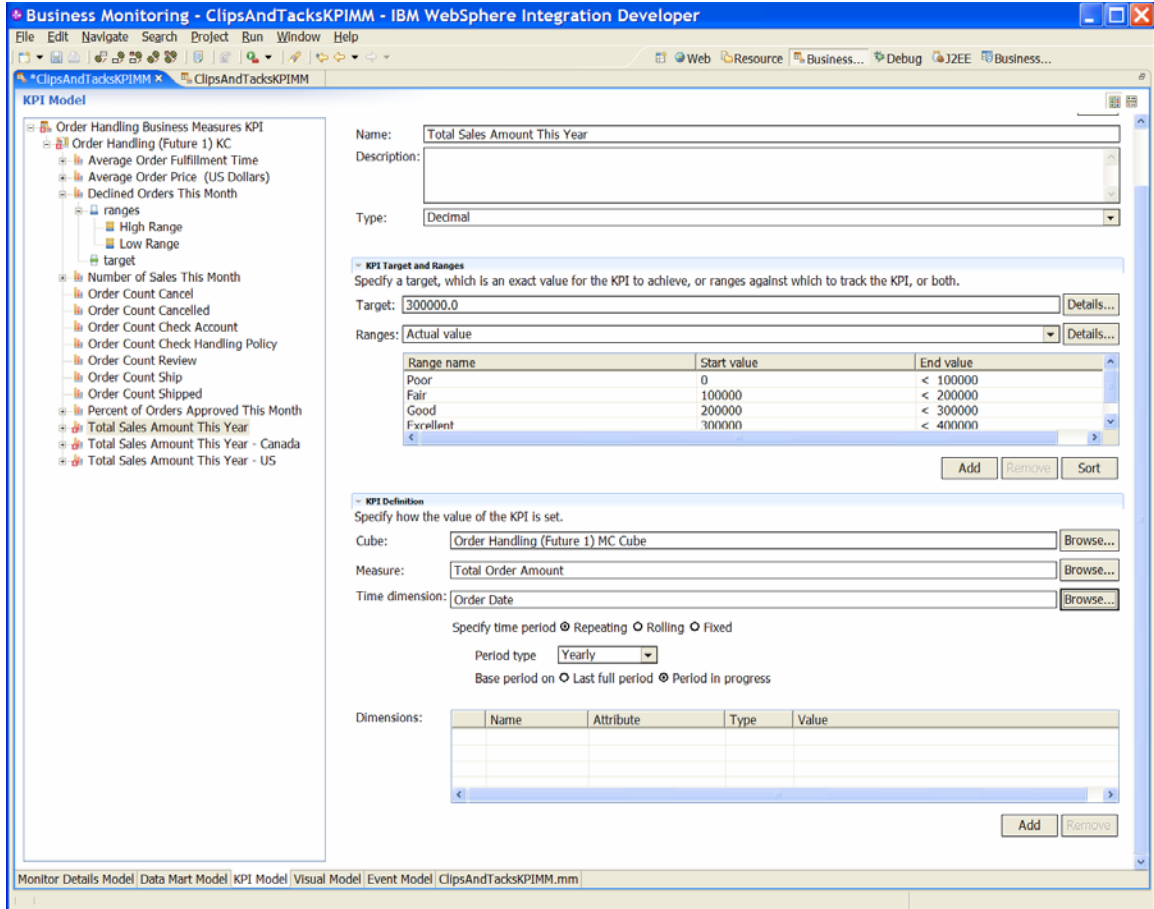
Figure 62. Add Completed string to is Approval Step Completed dimension





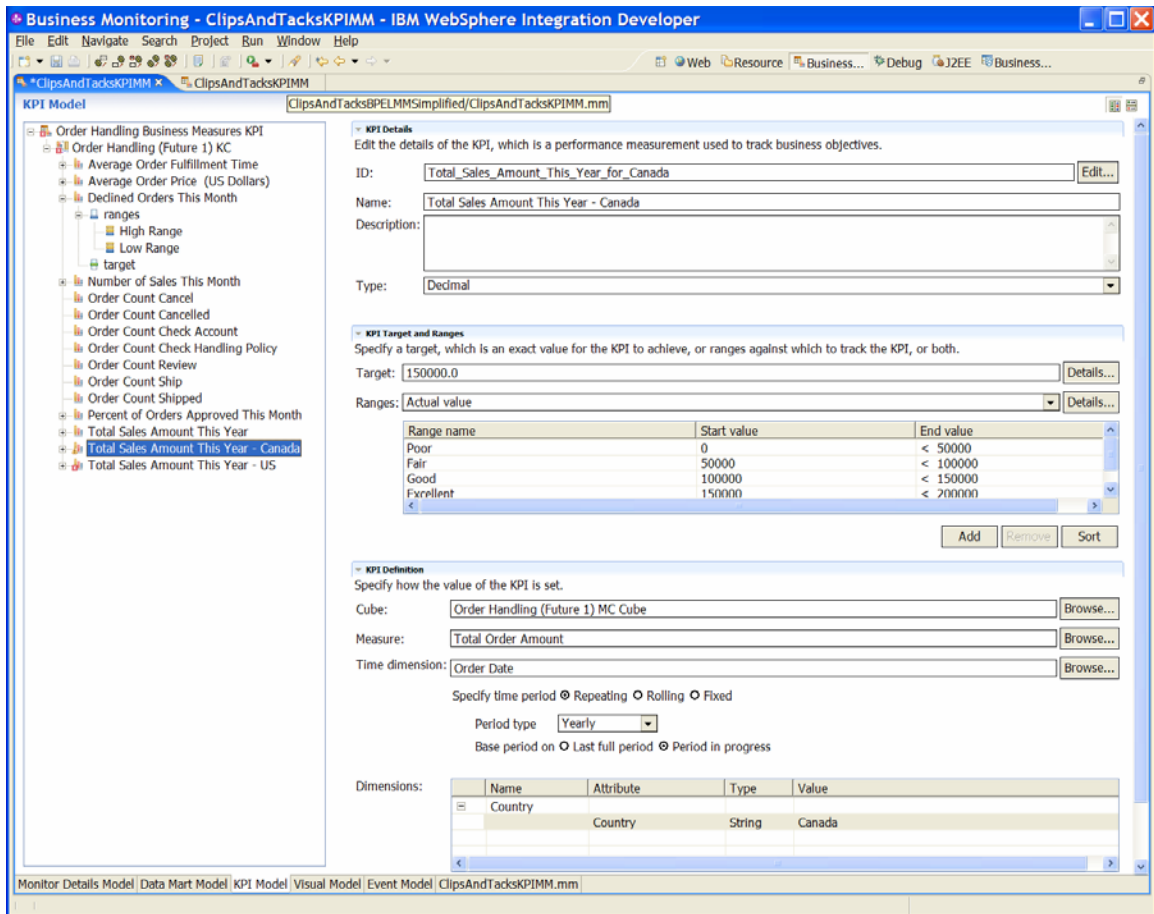
\_\_\_\_ 7. For the Total Sales Amount This Year, specify the Total Order Amount for the Measure and add the Order Date for the Time dimension, as seen here:

Figure 63. Specify Total Order Amount for the Measure



8. For the Total Sales Amount This Year – Canada KPI and the Total Sales Amount This Year – US KPI, specify the Total Order Amount for the **Measure** and add the **Order Date** for the **Time dimension**. Also, add the string Canada or US for the **Country** dimension. To avoid the warning about the ‘-’ character in the **ID** field, change the **ID** field. In this example, the **ID** field is changed to Total\_Sales\_Amount\_This\_Year\_for\_Canada. (You might need to change the ID and Name from Total\_Sales\_Amount\_This\_Year\_-\_Canada to Total\_Sales\_Amount\_This\_Year\_Ca. If the name is too long, it can cause problems during configuration of the KPI portlet.)

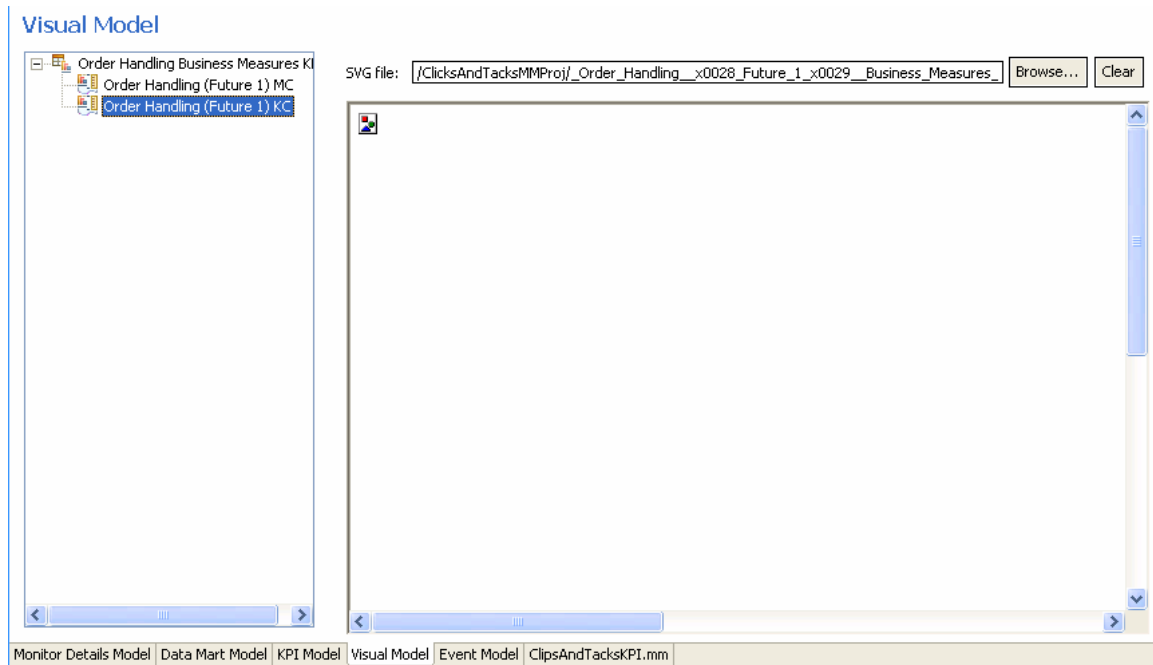
Figure 64. ID field changed to Total\_Sales\_Amount\_This\_Year\_for\_Canada



*Setting up the visual model for the KPI Model*

- \_\_\_ 1. Unzip the artifact file ClipsAndTacksSample1.zip to a directory, for example, c:\ClipsAndTacksSample1. There are two .svg file in the directory:
  - `_Order_Handling_x0028_Future_1_x0029_Business_Measures_KM_Order_Handling_x0028_Future_1_x0029_KC.svg` and
  - `_Order_Handling_x0028_Future_1_x0029_Business_Measures_MDM_Order_Handling_x0028_Future_1_x0029_MC.svg`
- \_\_\_ 2. Copy the two .svg file to the root of the monitor project **ClipsAndTacksSample**.
- \_\_\_ 3. Open the file **ClipsAndTackKPIMM.mm** with Monitor Model Editor and go to the **Visual Model** view.
- \_\_\_ 4. Select **Order Handling (Future 1) MC** context.
- \_\_\_ 5. Click **Browse** and select the file `_Order_Handling_x0028_Future_1_x0029_Business_Measures_MDM_Order_Handling_x0028_Future_1_x0029_MC.svg`
- \_\_\_ 6. Select **Order Handling (Future 1) KC** context.
- \_\_\_ 7. Click **Browse** and select the file `_Order_Handling_x0028_Future_1_x0029_Business_Measures_KM_Order_Handling_x0028_Future_1_x0029_KC.svg`

Figure 65. Copy .svg file from the artifacts zip file to your project



#### Part 4. Using the merged-model approach

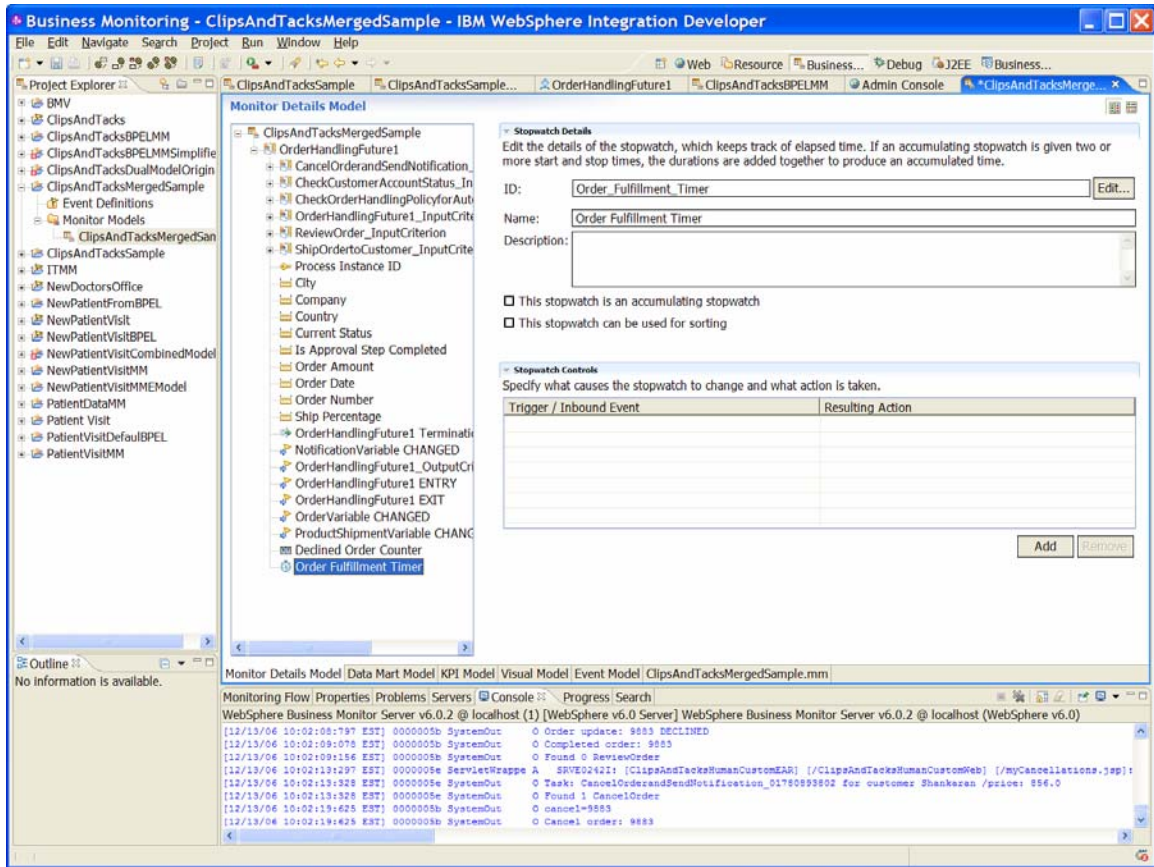
In the merged-model approach, you copy the KPI model into the BPEL model and use a single model. This section begins with a BPEL model that has the required metrics, events, triggers, and so on, but does not contain the outbound events to be sent to the KPI model. For the merged model, you still need to create the metrics that are needed for the KPIs, but you do not need the events that were used to send the data to the KPI model, because the KPIs are defined in this model. In this case, you are starting with the monitor model that is generated from the BPEL. Assuming that the monitor project is created from the BPEL Export, use the **File System ► Import** to import the BMV files into the same project that contains the BPEL Monitor model.

##### *Defining additional data needed for the KPI*

- \_\_\_\_\_ 1. First, add the metrics, counters, and stopwatches to the BPEL model to capture the data needed for the KPIs. Specifically, add the following metrics to the `OrderHandlingFuture1` monitor context:
  - **City**: type string, maximum length of 256, default value of 'Durham'
  - **Company**: string, maximum length of 256, no default value
  - **Country**: string, maximum length of 256, default value of 'US'
  - **Current Status**: string, maximum length of 256, no default value
  - **Is Approval Step Completed?**: id field set to `isApprovalStepCompleted`, string, maximum length of 256, default value of 'Not Completed'
  - **Order Amount**: decimal, default value of 0.
  - **Order Date**: type Date, no default value
  - **Order Number**: type Integer, no default value
  - **Ship Percentage**: type Integer, default value of 0
  
- \_\_\_\_\_ 2. Next, add the **Declined Order Counter** and the **Order Fulfillment Timer** to the `OrderHandlingFuture1` monitor context.

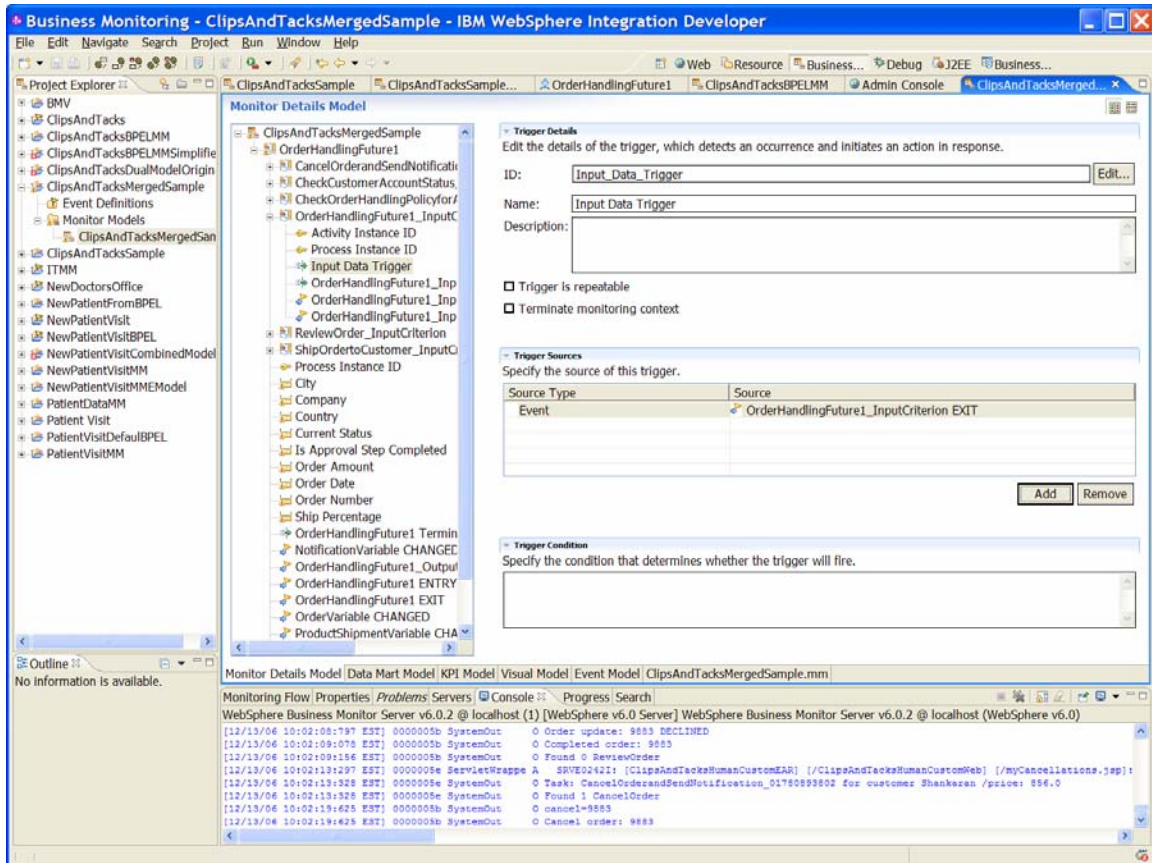
The following figure shows the BPEL model with the metrics, counter, and stopwatch added:

Figure 66. BPEL model with metrics, counter, and stopwatch



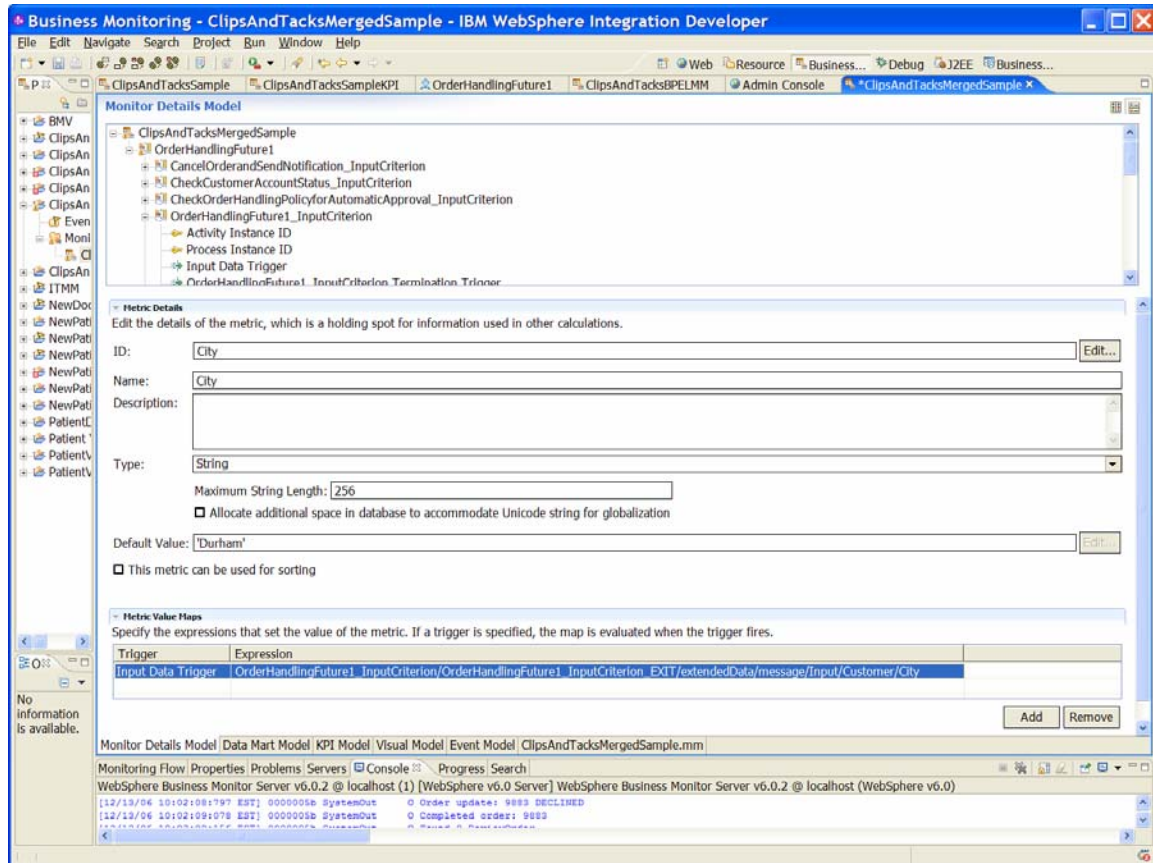
3. Next, add the triggers that cause the metrics, counters, and stopwatches to get updated. First, add a trigger that is associated with the **OrderHandlingFuture1\_InputCriterion EXIT** inbound event. This inbound event contains the data entered in the BPEL process. This trigger is used to set the value in City, Company, Country, Order Amount, and Order Number. First create the trigger, **Input Data Trigger**, in the **OrderHandlingFuture1\_InputCriterion** context and add a trigger source for the **OrderHandlingFuture1\_InputCriterion EXIT** event. The new trigger is shown here:

Figure 67. Add triggers that update metric, counter, and stopwatch



- \_\_\_\_\_ 4. Next, for the metrics that update when this trigger fires, add a map on the associated metric that sets the metric from the value in the inbound event. Shown here is the map for the **City** metric (shown in vertical orientation to show the complete map expression):

Figure 68. Map for City metric



- \_\_\_\_\_ 5. Add the map for the Company, Country, Order Amount, and Order Number that get set by this trigger. The Order Amount is set to the **Total Price** field from the inbound event.
- \_\_\_\_\_ 6. Next, add the triggers that set the **Current Status** metric. The **Current Status** metric is set to one the following values (depending upon the location of the order in the order process): Started, Cancelled, Shipped, Cancel Order, Check Account, Checked Account, Review Order, Reviewed, Check Order Handling Policy, Checked Order Handling Policy, and Ship Order.

Following is the list of the triggers that are used to set the status. The termination triggers are generated for you in the BPEL model, but you need to add the other triggers. For triggers that need to be added, the monitor context where the trigger is to be added, and the event source for the trigger are specified here:

- New Order Trigger
  - **Monitor Context:** OrderHandling Future1
  - **Event Source:** OrderHandlingFuture1 ENTRY inbound event
  - This trigger needs to be added.
- CancelOrderandSendNotification\_InputCriterion Termination Trigger
  - **Monitor Context:** CancelOrderandSendNotification\_InputCriterion
  - This trigger should already exist in the model because termination triggers are automatically created in the default BPEL model.
- ShipOrdertoCustomer\_InputCriterion Termination Trigger
  - **Monitor Context:** ShipOrdertoCustomerInputCriterion
  - This trigger should already exist.
- Cancel Order Started Trigger
  - **Monitor Context:** CancelOrderandSendNotification\_InputCriterion
  - **Event Source:** CancelOrderandSendNotification\_InputCriterion ENTRY inbound event
  - This trigger needs to be created. The specified inbound event should be in the generated model.
- Check Account Status Started Trigger
  - **Monitor Context:** CheckCustomerAccountStatus\_InputCriterion
  - **Event Source:** CheckCustomerAccountStatus\_InputCriterion ENTRY inbound event
  - This trigger needs to be created. The specified inbound event should already exist in the specified model.
- CheckCustomerAccountStatus\_InputCriterion Termination Trigger
  - **Monitor Context:** CheckCustomerAccountStatus\_InputCriterion
  - This trigger should already exist.
- Review Order Started Trigger
  - **Monitor Context:** ReviewOrder\_InputCriterion
  - **Event Source:** ReviewOrder\_InputCriterion ENTRY inbound event
  - This trigger needs to be created.
- The specified inbound event should already exist in the generated model.



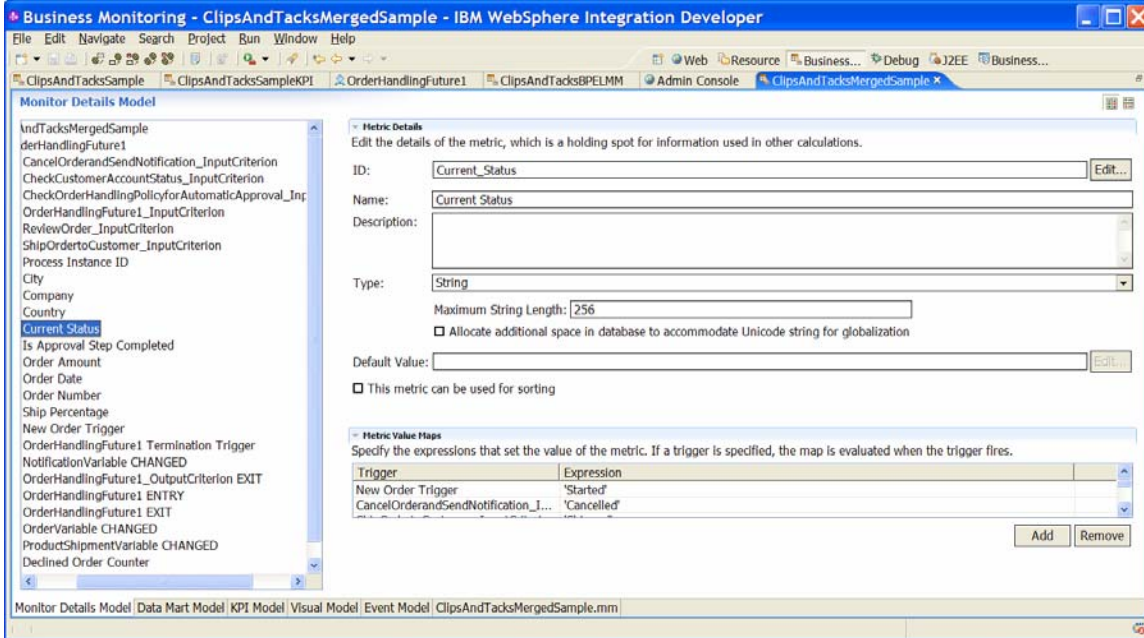
- Review Order\_InputCriterion Termination Trigger
  - **Monitor Context:** ReviewOrder\_InputCriterion MC
  - This trigger should already exist.
- Check Order Handling Policy Started Trigger
  - **Monitor Context:** in the CheckOrderHandlingPolicyforAutomaticApproval\_InputCriterion MC
  - **Event Source:** CheckOrderHandlingPolicyforAutomaticApproval\_InputCriterion ENTRY inbound event
  - This trigger needs to be created. The specified inbound event should already exist in the generated model.
- CheckOrderHandlingPolicyforAutomaticApproval\_InputCriterion Termination Trigger
  - **Monitor Context:** CheckOrderHandlingPolicyforAutomaticApproval\_InputCriterion MC
  - This trigger should already exist.
- Ship Order Started Trigger
  - **Monitor Context:** ShipOrdertoCustomer\_InputCriterion MC
  - **Event Source:** ShipOrdertoCustomer\_InputCriterion ENTRY event
  - This trigger needs to be created.

\_\_\_\_ 7. After adding the new triggers, add the maps to the Current Status metric to set the correct value for each trigger. Listed here are each trigger and the values that should be specified for the expression when that triggers fires:

- New Order Trigger – Started
- CancelOrderandSendNotification\_InputCriterion Termination Trigger – Cancelled
- ShipOrdertoCustomer\_InputCriterion Termination Trigger – Shipped
- Cancel Order Started Trigger – Cancel Order
- Check Account Status Started Trigger – Check Account
- CheckCustomerAccountStatus\_InputCriterion Termination Trigger – Checked Account
- Review Order Started Trigger – Review Order
- ReviewOrder\_InputCriterion Termination Trigger – Reviewed
- Check Order Handling Policy Started Trigger – Check Order Handling Policy
- CheckOrderHandlingPolicyforAutomaticApproval\_InputCriterion Termination Trigger – Checked Order Handling Policy
- Ship Order Started Trigger – Ship Order

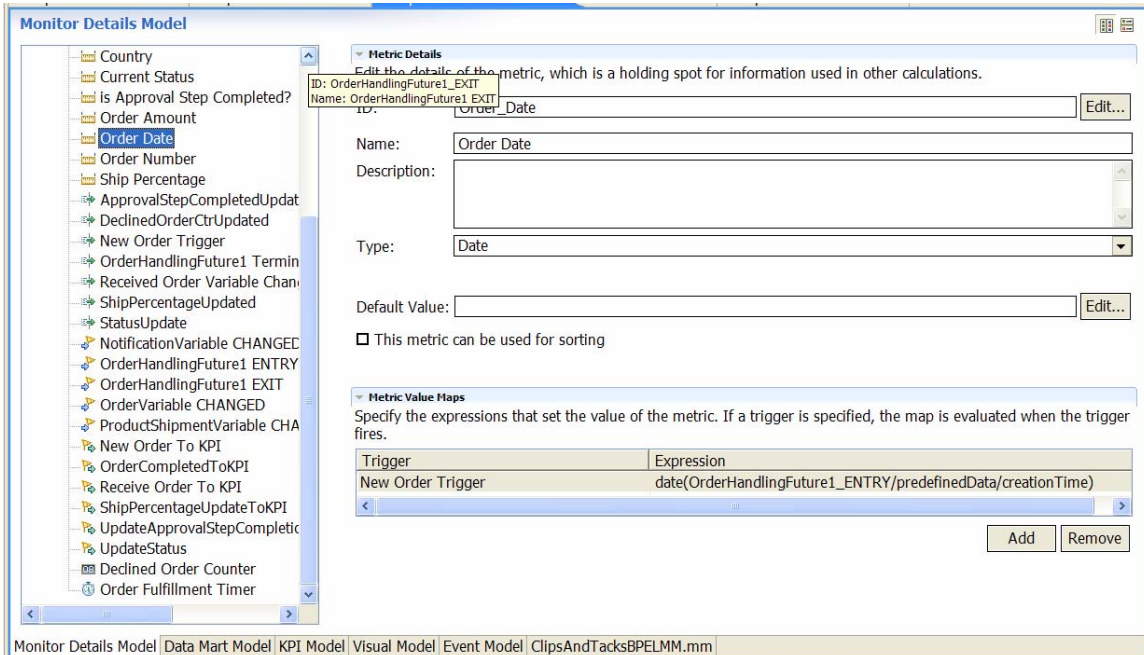
Shown here is the **Current Status** metric with the Metric Value Maps displayed for the **New Order Trigger** and the **CancelOrderandSendNotification\_InputCriterion Termination Trigger**:

Figure 69. Current Status metric with Metric Value Maps



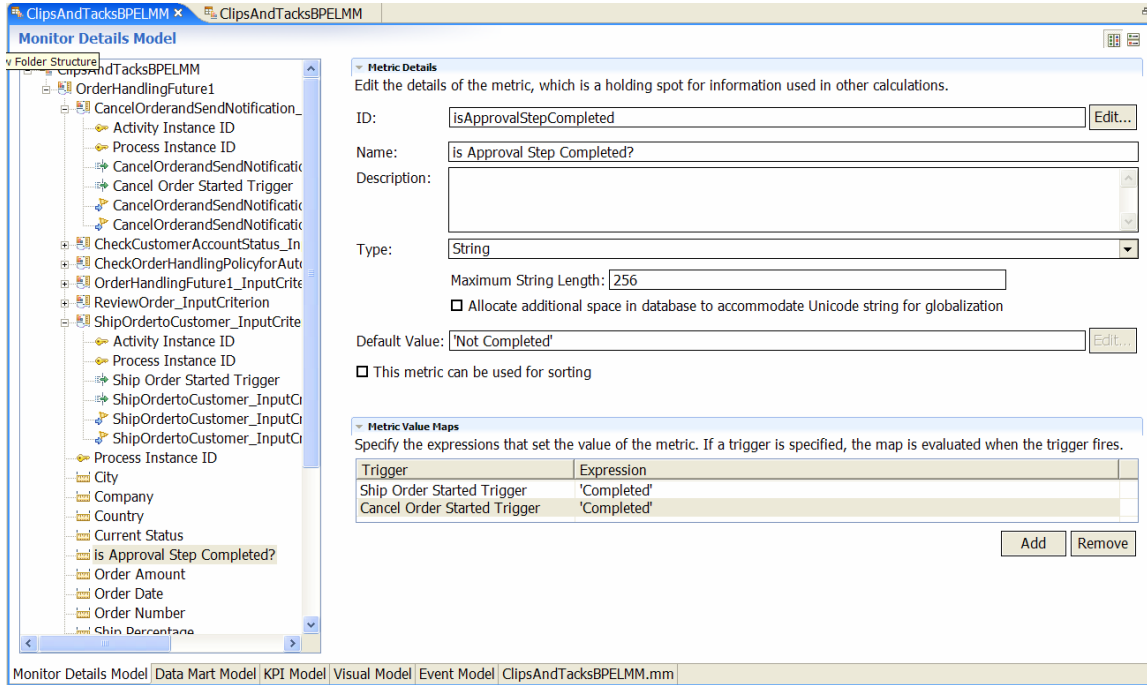
8. Now the map for the Order Date needs to be set, use the date function with the creationTime field in the **OrderHandlingFuture1 ENTRY** inbound event, as shown here (the **creationTime** is a dateTime field that must be converted to a Date type before it can be placed in the **Order Date** metric, which is of type **Date**):

Figure 70. Setting Order Date



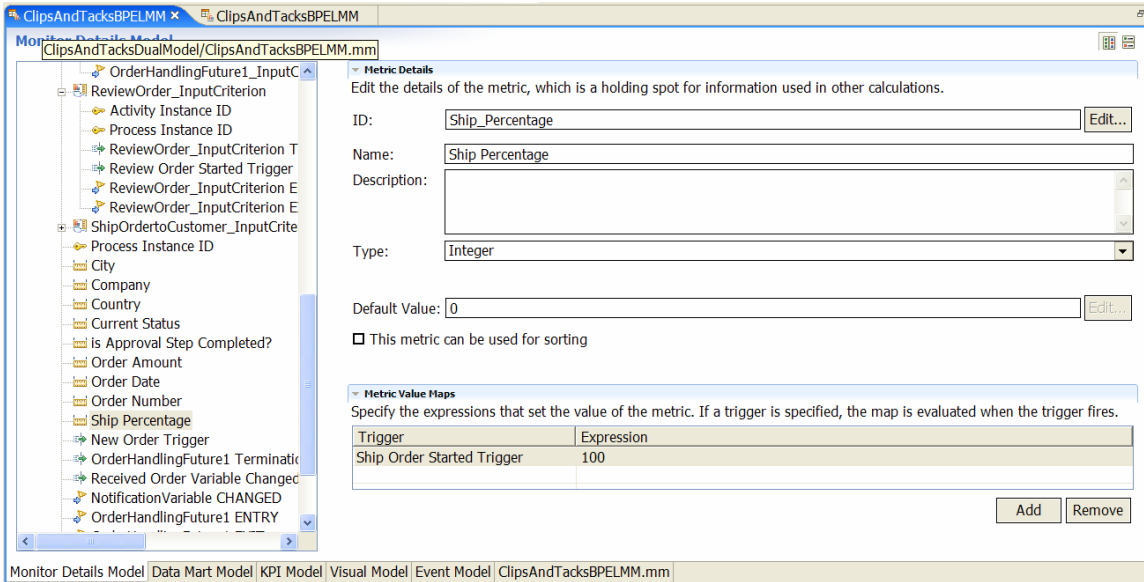
9. The **Is Approval Step Completed** metric is also set by triggers, so add the maps to set this value next. This metric is set to **Completed** when either the **Ship Order Started Trigger** or the **Cancel Order Started Trigger** is fired. Shown here is this metric with the added metric maps for the triggers:

Figure 71. Is Approval Step Completed metric with added metric maps for triggers



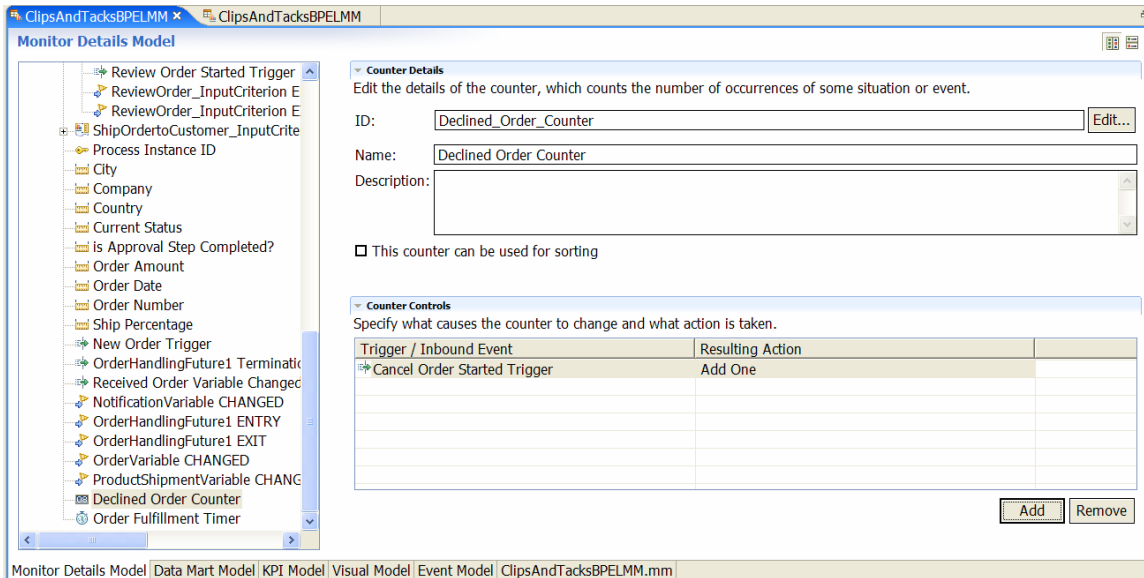
The **Ship Percentage** metric is set to 100 when the **Ship Order Started Trigger** is fired. Shown here is the **Ship Percentage** metric with its map:

Figure 72. Ship Percentage metric with its map



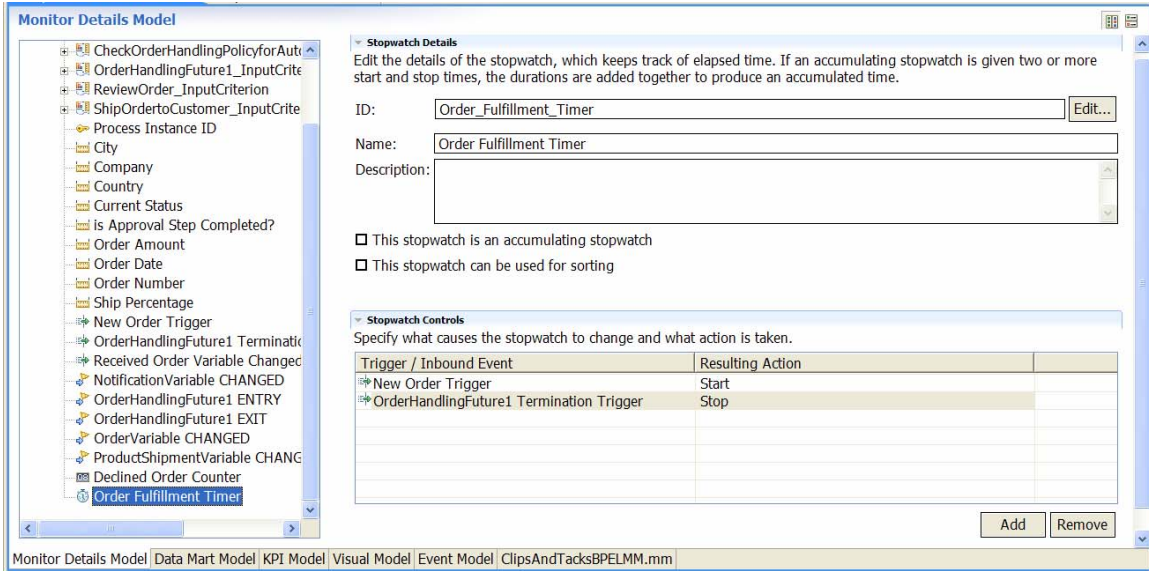
10. Next, add the trigger for the **Declined Order Counter**. Whenever the **Cancel Order Started Trigger** is fired, the **Declined Order Counter trigger** should be incremented. This is illustrated in the following example:

Figure 73. Add trigger for the Declined Order Counter



11. Next, add the triggers that start and stop the **Order Fulfillment Timer**. This timer is started when the order is received, so the **New Order Trigger** starts the timer and the timer is stopped with the process is terminated, so the **OrderHandlingFuture1 Termination Trigger** is used to stop the timer. This is seen in the following figure:

Figure 74. OrderHandlingFuture1 Termination Trigger used to stop the timer



12. At this point, in the BPEL model, begin the merge from the KPI model. To merge the KPI model, select the model that is exported from BMV that contains the KPI model. As shown in this example, click the tab named **modelName.mm** and find the section of the .mm file that contains the following line:  
`<kpiModel id="KM">`

Figure 75. Find this section of the .mm file



- \_\_\_\_\_ 13. Now copy the text between the `<kpiModel id="KM">` and the end tag `</kpiModel>` and paste this text into the BPEL model after the `</monitorDetailsModel>` tag in the XML view of the merged model.

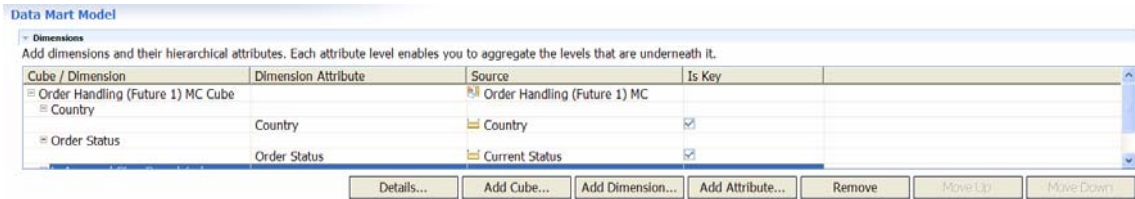
### *Setting up the Data Mart model*

Now update the Data Mart model to set up the dimensions, facts, and measures.

- \_\_\_\_\_ 1. First, set these dimensions on the Order Handling cube. The dimension should already be specified, but the dimension attributes and dimension sources are missing. These dimensions are used in the KPI model, so they need to be completed here.
  - **Country:** for Dimension Attribute and Source, specify Country; select the Is Key check box.
  - **Order Status:** for Dimension Attribute, specify Order Status; for Source, specify Current Status; select the Is Key check box.
  - **Order Date:** for Dimension Attribute and Source, specify Order Date; select the Is Key check box.
  - **Is Approval Step Completed:** for Dimension Attribute and Source, specify Is Approval Step Completed; select the Is Key check box.

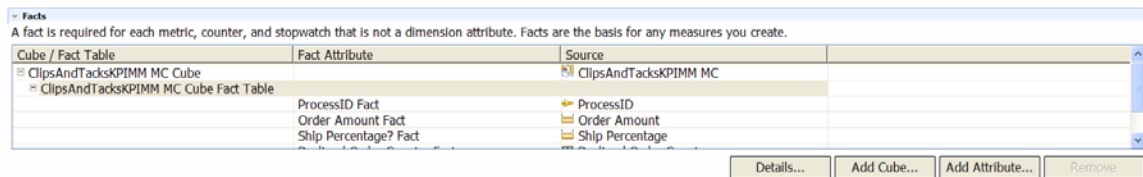
Shown here is the Data Mart model page showing the details for the **Country and Order Status dimensions**:

Figure 76. Data Mart model page details for Country and Order Status dimensions



- Now, fix the facts. The following facts should be in the fact table: **Process ID, Order Amount, Ship Percentage, Declined Order Counter, and Order Fulfillment Timer**. For details, see Setting up the Data Mart model on page 58. Some of these are shown here:

Figure 77. Process ID, Order Amount, Ship Percentage in fact table



- Now, ensure the measures are correct. These measures need to be created in the OrderHandlingFuture1 cube:

- Total Order Amount** – Source Attribute is Order Amount Fact and the Aggregation Function is Sum.
- Average Order Amount** – Source Attribute is Order Amount Fact and the Aggregation Function is Average.
- Average Order Fulfillment Time** – Source Attribute is Order Fulfillment Timer Fact and the Aggregation Function is Average
- Number of Orders** – Source Attribute is ProcessID Fact and the Aggregation Function is Count.
- Ship Percentage** – Source Attribute is Ship Percentage Fact and the Aggregation Function is Average.
- Declined Orders** – Source Attribute is Declined Order Counter Fact and the Aggregation Function is Sum.

The following figure shows the Measures:

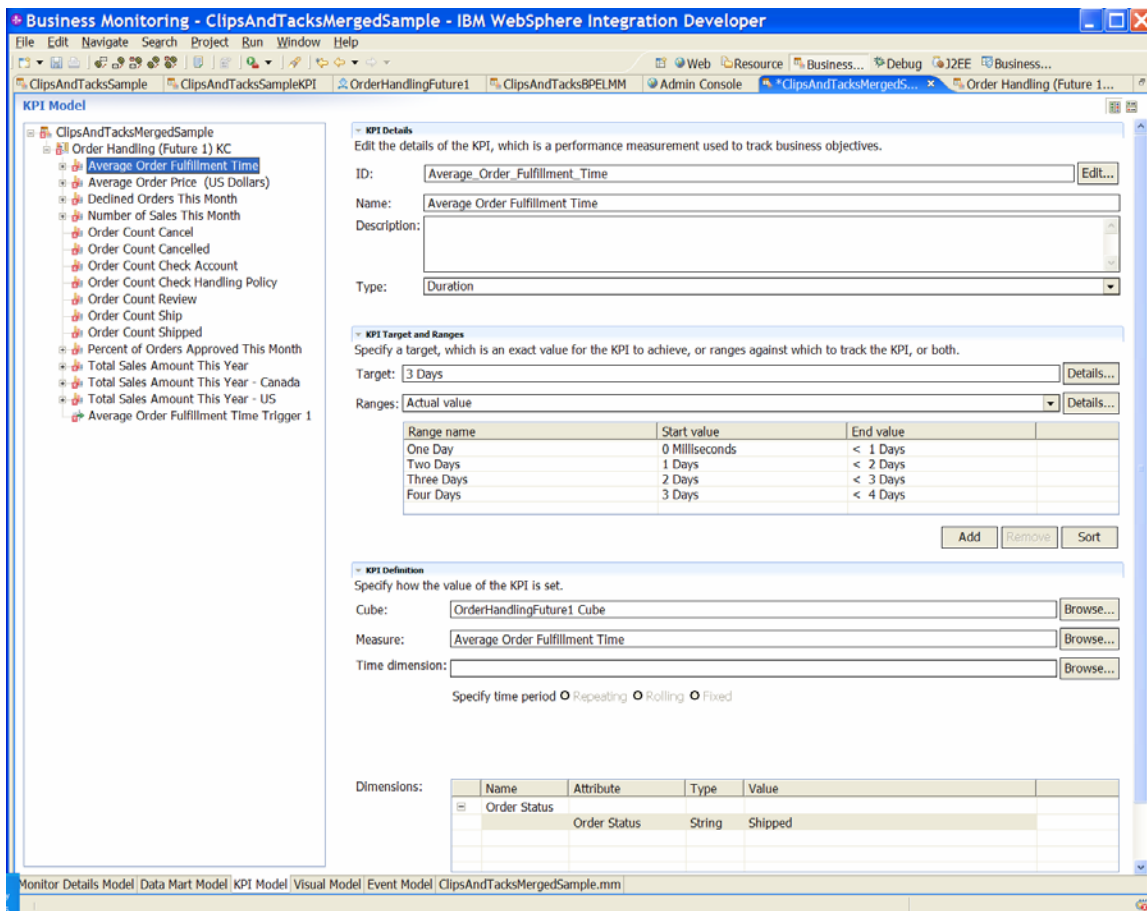
Figure 78. Measures in OrderHandlingFuture1 cube

| Cube                              | Measure                        | Source                       | Aggregation Function |
|-----------------------------------|--------------------------------|------------------------------|----------------------|
| Order Handling (Future 1) MC Cube | Total Order Amount             | Order Handling (Future 1) MC | Sum                  |
|                                   | Average Order Amount           | Order Amount Fact            | Average              |
|                                   | Average Order Fulfillment Time | Order Fulfillment Timer Fact | Average              |
|                                   | Number of Orders               | ProcessID Fact               | Count                |

*Setting up the KPI model*

- Next, go to the KPI model view and complete the KPI model. For the **KPI Average Order Fulfillment Time**, the **Cube, Order Handling Future1** and the **Measure, Average Order Fulfillment Time** needs to be specified. Also, the dimension value that is used needs to be specified. Add the value **Shipped** to the **Order Status Dimension**, as illustrated below:

Figure 79. Add value Shipped to Order Status Dimension



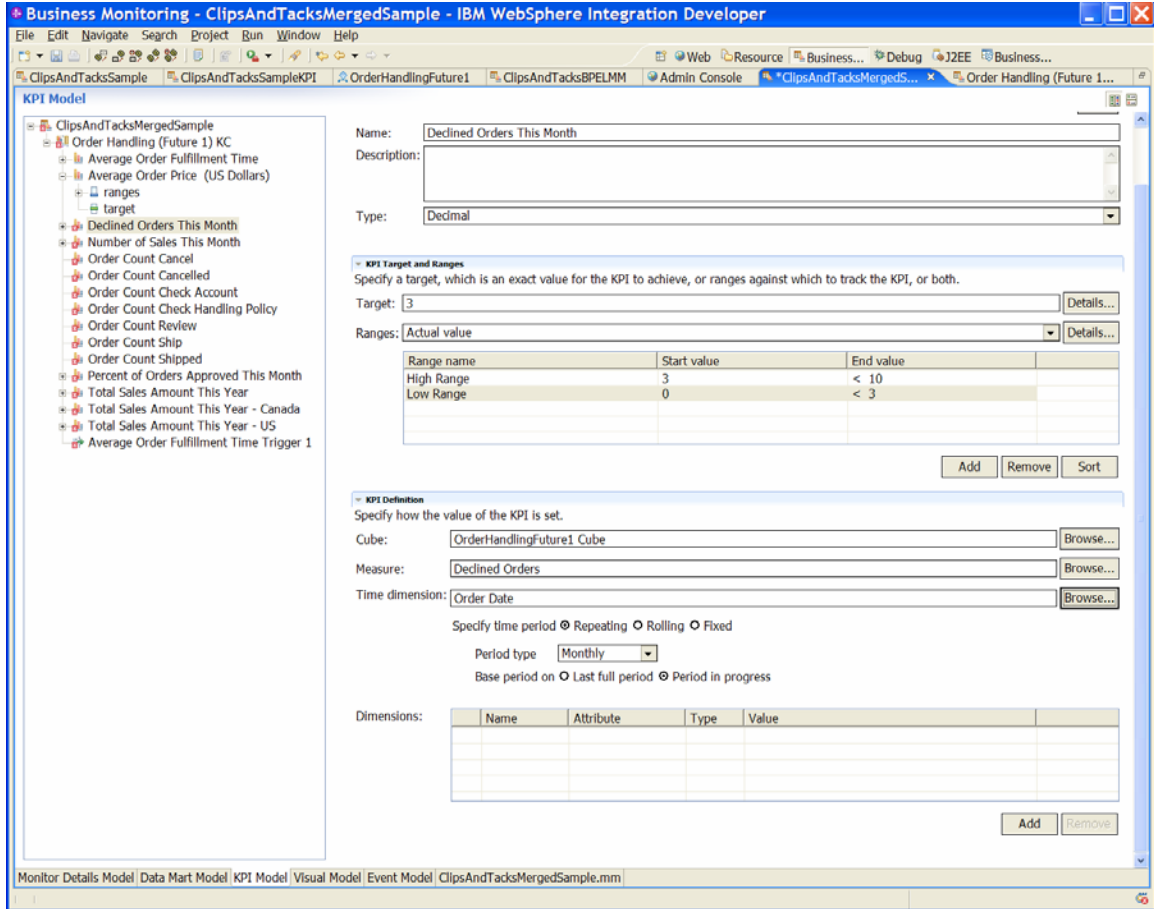
- For the **Average Order Price KPI**, specify the **OrderHandlingFuture1** cube and specify a measure of **Average Order Amount**. (It may also be necessary to use the XML editor to delete and extraneous `<dimension/>` tag in the XML. Make **ID** and **Name** as short as possible.



For example, you can change the ID to Average\_Order\_Price\_US\_Dollars.

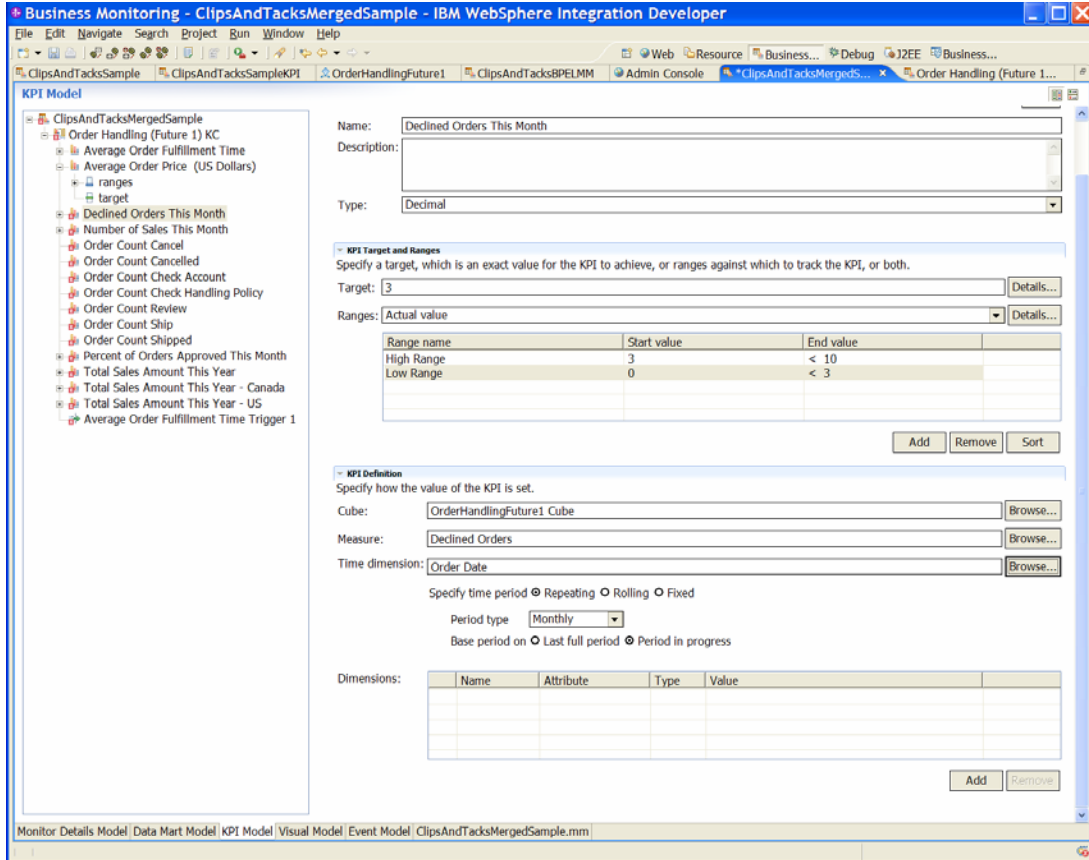
- \_\_\_\_\_ 3. For the **Declined Orders This Month**, specify the **OrderHandlingFuture1 Cube**, the **Declined Order Measure** and the **Order Date Time dimension**. Also make sure the type of **Decimal** is specified. You may also need to specify **Target** of 3, a **High Range** from 3 to 10 and a **Low Range** from 0 to 3. When entering the **Ranges**, a **Name** for each range attribute should also be specified.

Figure 80. Specify a Name for each range attribute



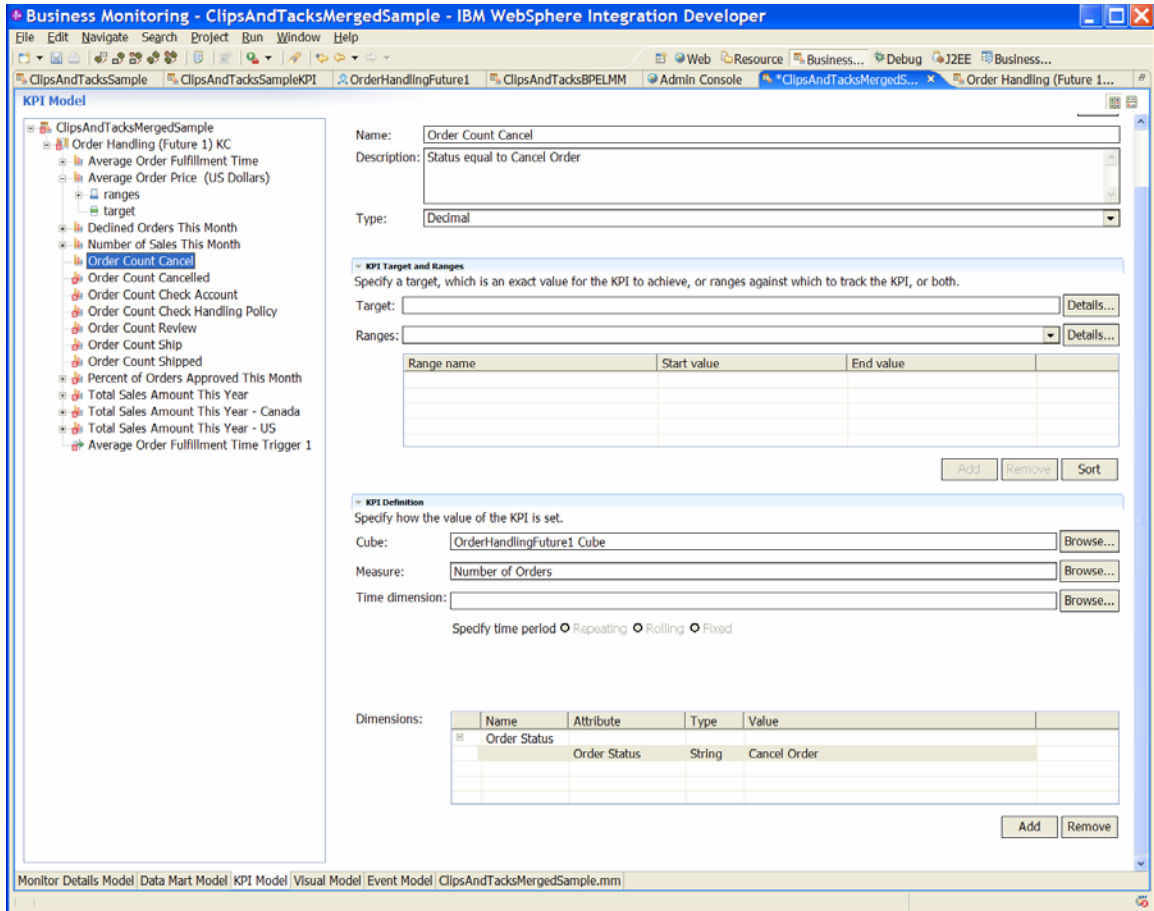
4. For the **Number of Sales This Month KPI**, add the **OrderHandlingFuture1 Cube**, and set **Measure** to **Number of Orders** and **Time dimension** to **Order Date** as shown:

Figure 81. Set Measure to Number of Orders and Time dimension to Order Date



- \_\_\_\_ 5. For the **Order Count Cancel KPI**, specify the **OrderHandlingFuture1** Cube and set **Measure** to **Number of Orders**. Next add the **Cancel Order** value string to the **Order Status** dimension and set **Type** to **Decimal**, as shown here:

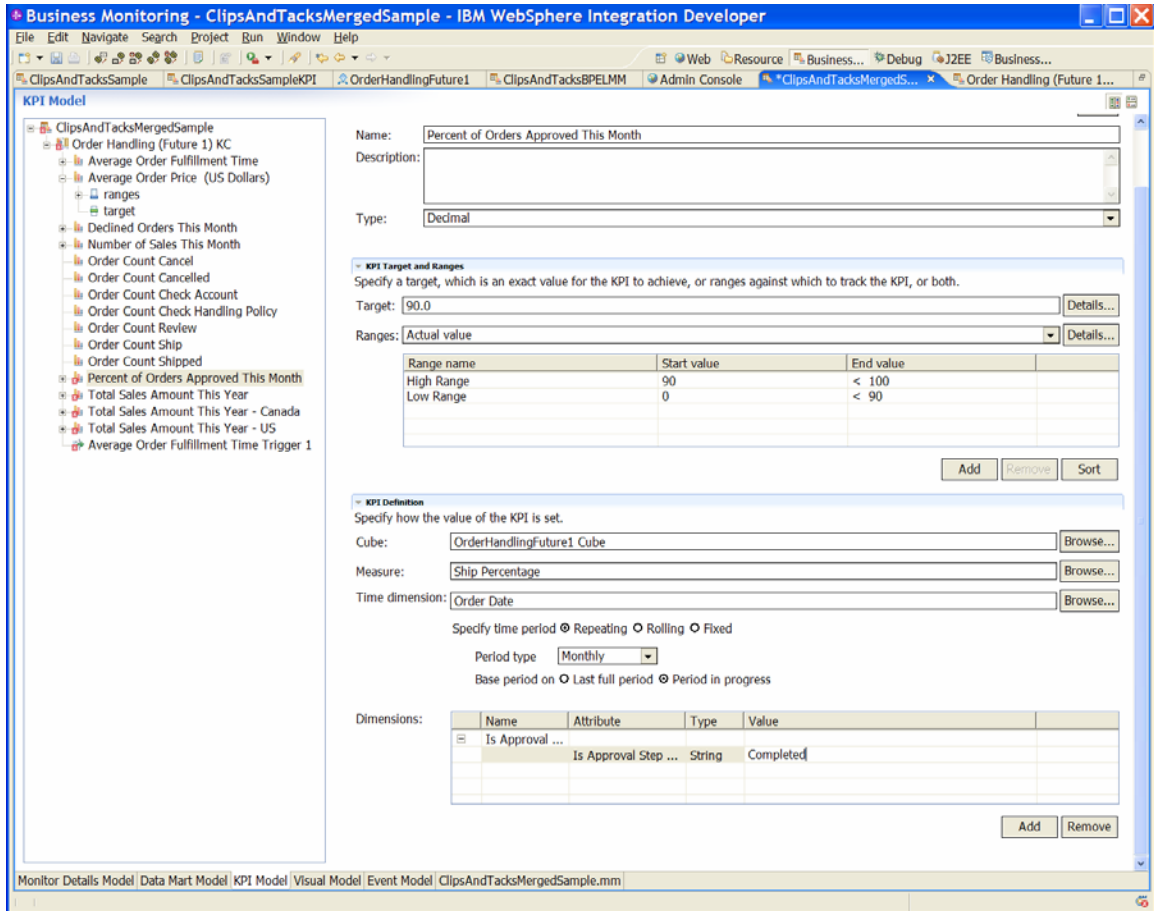
Figure 82. Set Type to Decimal



- \_\_\_\_ 6. Now, make sure each of these KPIs has Type set to Decimal, has Cube set to OrderHandlingFuture1, has Measure set to the Number of Order, and this value in the Order Status dimension check:
- Order Count Cancelled - Cancelled
  - Order Count Check Account – Check Account
  - Order Count Check Handling Policy – Check Order Handling Policy
  - Order Count Review – Review Order
  - Order Count Ship – Ship Order
  - Order Count Shipped – Shipped

- \_\_\_\_\_ 7. For the **Percent of Order Approved This Month KPI**, add the **OrderHandlingFuture1 Cube**, set **Measure** to **Ship Percentage**, set **Time Dimension** to **Order Date**, and add the **Completed** string to the **is Approval Step Completed** dimension. This is shown here:

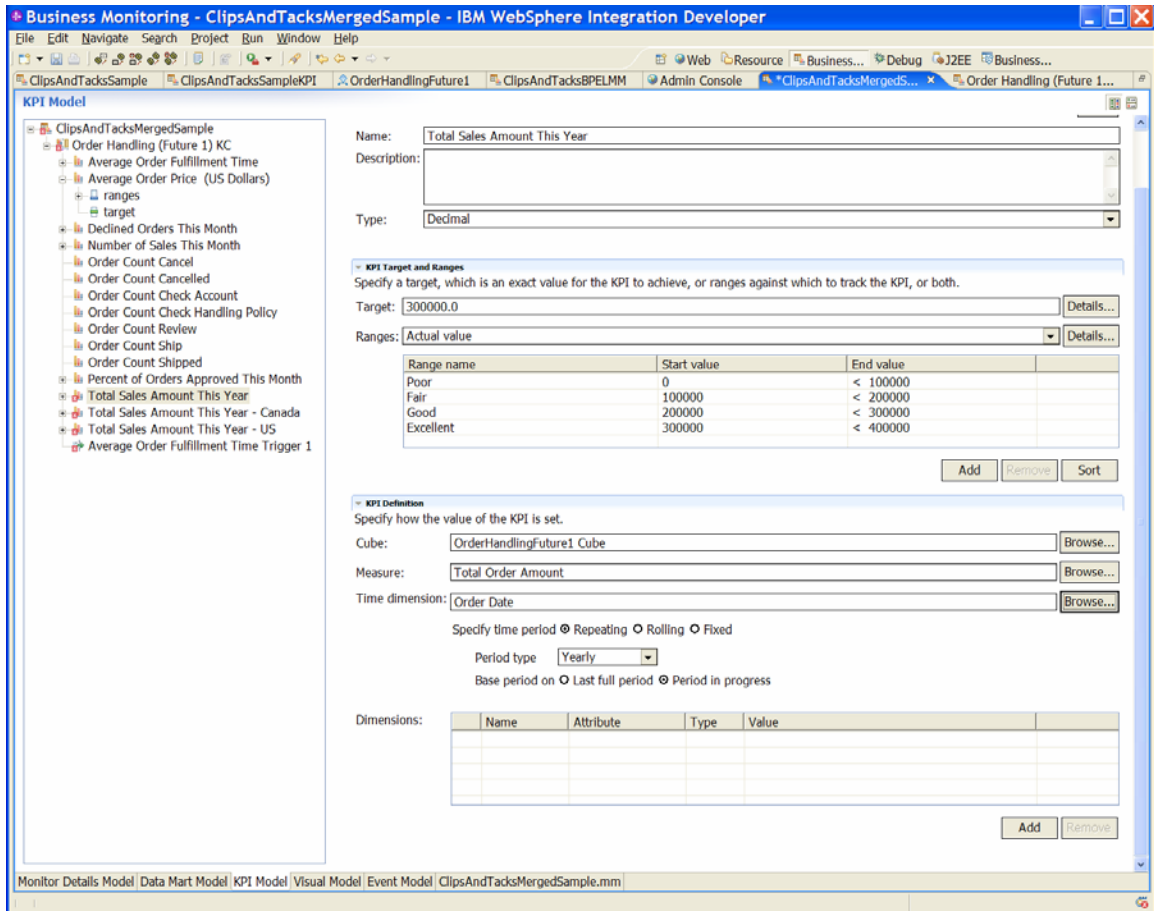
Figure 83. Add Completed string to is Approval Step Completed dimension



- \_\_\_\_\_ 8. Change the ID and Name values to be as short as possible. For example, you can change the ID to **Perc\_of\_Orders\_Approved\_Month** and the Name to **Perc of Orders Approve Month**.

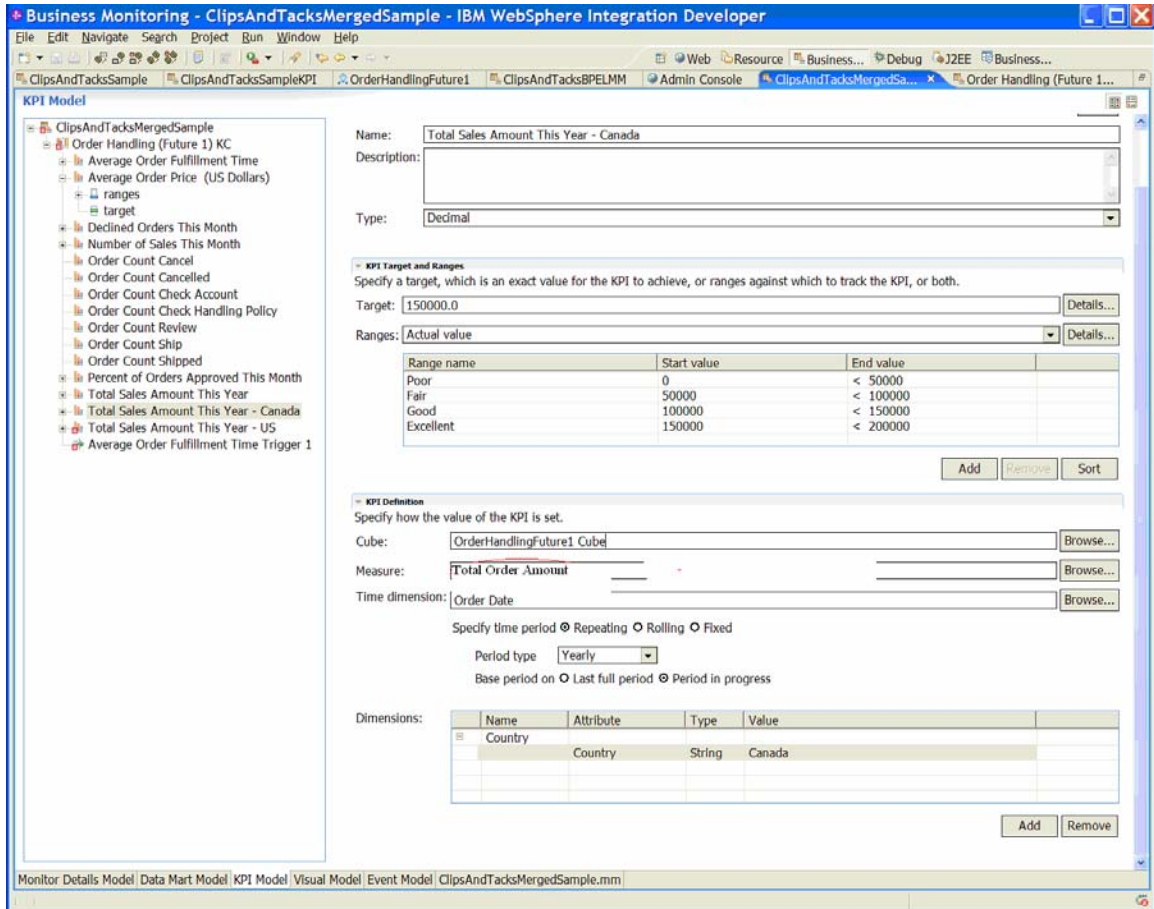
9. For the **Total Sales Amount This Year**, specify the **OrderHandlingFuture1** cube, set **Measure** to **Total Order Amount** and **Time dimension** to **Order Date**, as seen here:

Figure 84. Set Measure to Total Order Amount and Time dimension to Order Date



- \_\_\_\_\_ 10. For the **Total Sales Amount This Year – Canada KPI** and the **Total Sales Amount This Year – US KPI**, specify the **OrderHandlingFuture1 Cube**, set **Measure** to **Total Order Amount** and **Time dimension** to **Order Date**. Also, add the string **Canada** or **US** for the **Country dimension**. The **Canada KPI** is shown here:

Figure 85. Add the string Canada or US for the Country dimension



- \_\_\_\_\_ 11. Make **ID** and **Name** as short as possible. For example, to **Total Sales Amount This Year – Canada KPI**, you can change the **ID** to **Total\_Sales\_Amount\_This\_Year\_Ca** and the **Name** to **Total Sales Amount This Year Ca**.

- \_\_\_\_\_ 12. Finally, delete the **Average Order Fulfillment Time Trigger 1**. This completes the merging of the KPI model and the BPEL model.

### Part 5. Deciding which approach to use

The dual-model approach is certainly the more complex approach. The data required in the KPI model must be duplicated in the BPEL model, and the events needed to keep the KPI model up-to-date must be created and debugged. The dual-model approach requires more upfront development time, including the creation and deployment of multiple models with the associated databases. The dual-model approach results in performance less than the merged-model approach.

Maintenance is another consideration for deciding which approach to use.

- If the data that is being generated is relatively static, but the BPEL process is changing, it might be easier from a maintenance standpoint to use the dual-model approach. If the BPEL model changes, you can make the required changes to the BPEL, but you might not have to make any changes to the KPI model.
- Otherwise, the simpler maintenance for the merged model makes it the better approach to use for most situations.

### Part 6. How to run the sample

Follow these steps to run data through the process.

- \_\_\_ 1. Start Monitor Server, Business Process Server and Dashboard Server.
- \_\_\_ 2. After deploying the EAR files, run these BPEL process steps.
  - \_\_\_ a. Create the database and tables:  
<http://localhost:9082/ClipsAndTacksFrontWeb/CreateDatabaseServlet>
  - \_\_\_ b. Place orders:  
<http://localhost:9082/ClipsAndTacksFrontWeb>
  - \_\_\_ c. Process orders:  
<http://localhost:9082/ClipsAndTacksHumanCustomWeb/>.  
Run the custom human task applications to process the human tasks. You can review orders and approve or decline them.

The server name and port in the URL may need to be changed according to your server configuration.

For detailed information, see the document for the sample scenario entitled **Generating a Monitor Model for a BPEL process**.





© Copyright IBM Corporation 2007

IBM Corporation  
Software Group  
Route 100  
Somers, NY 10589  
U.S.A.

Produced in the United States of America  
06-07

All Rights Reserved

ALPHABLOX is a registered trademark of Alphablox Corporation in the United States, other countries, or both.

(See this URL for details:

<http://www.ibm.com/legal/copytrade.shtml#section-alphablox>)

Cloudscape, DB2, DB2 Universal Database, IBM, the IBM logo and WebSphere are trademarks of International

Business Machines Corporation in the United States, other countries or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.

Other company, product and service names may be trademarks or service marks of others.