WebSphere Business Monitor V6.1 - Clips and Tacks Business Activity Monitoring using XSD event definition

What this exercise is about

The objective of this lab is to show you how to build a monitor model using XSD event definitions in Rational Application Developer or WebSphere Integration Developer, deploy it to WebSphere Business Monitor and then view your monitored data on the Monitor dashboards.

This lab will show you Business Activity Monitoring (BAM) which involves event based monitoring. With BAM, you can have your monitored application running anywhere, and submitting events to the Monitor server so that you can view monitored data. Typically, you identify the events that are created by the application, and then create a monitor model that represents the monitored data that you want to collect from the events.

Changes from the previous version

This lab has been modified significantly from the previous 6.0.2 release. Here are the major changes:

- In the monitor model, you can reference other KPIs when defining the calculation for a given KPI.
- KPIs are now based on a metric and aggregation function, and not based on cube measures, so the dimensional model is now simplified.
- Deploying the model has been dramatically simplified, so this is a very short process now.
- LDAP is not required, so now you can test alerts without installing LDAP.
- Portal is no longer a requirement, so configuration of a dashboard is simplified.

Lab requirements

List of system and software required for the student to complete the lab:

- Rational Application Developer V7.0 or WebSphere Integration Developer V6.1.
- WebSphere Business Monitor V6.1 – Toolkit Installation including the Monitor Model editor and Monitor Server

What you should be able to do

At the end of this lab you should be able to:

- Use Rational Application Developer or WebSphere Integration Developer to create the monitor model, and deploy it to the server.
- Use a sample program to create events that represent events from your process.
- Define a dashboard to view monitored data.

Introduction

In this lab you will learn how to develop and deploy a model from end to end using Rational Application Developer or WebSphere Integration Developer and WebSphere Business Monitor. This lab will show you the basic procedure for creating and implementing a simple model.

The model used in this lab is **ClipsAndTacks** model which is also used in IBM Redbook **SG247148** – **Business Process Management: Modeling through Monitoring Using WebSphere V6.0.2 Products**. In the Redbook, there are several models that are referenced, but in this lab you will be using the Future 1 Process. In the Redbook, the process is implemented as a BPEL process running in WebSphere Process Server. However this lab will demonstrate **Business Activity Monitoring** which can monitor events from any source. So, the lab will simulate the running of the process, by creating events that could have been sourced from anywhere, a Java EE application or BPEL, or any other source. These events are in the form of Common Base Events, which is an OASIS standard for common event format. These events are used by the Common Event Infrastructure (CEI) in WebSphere Process Server.

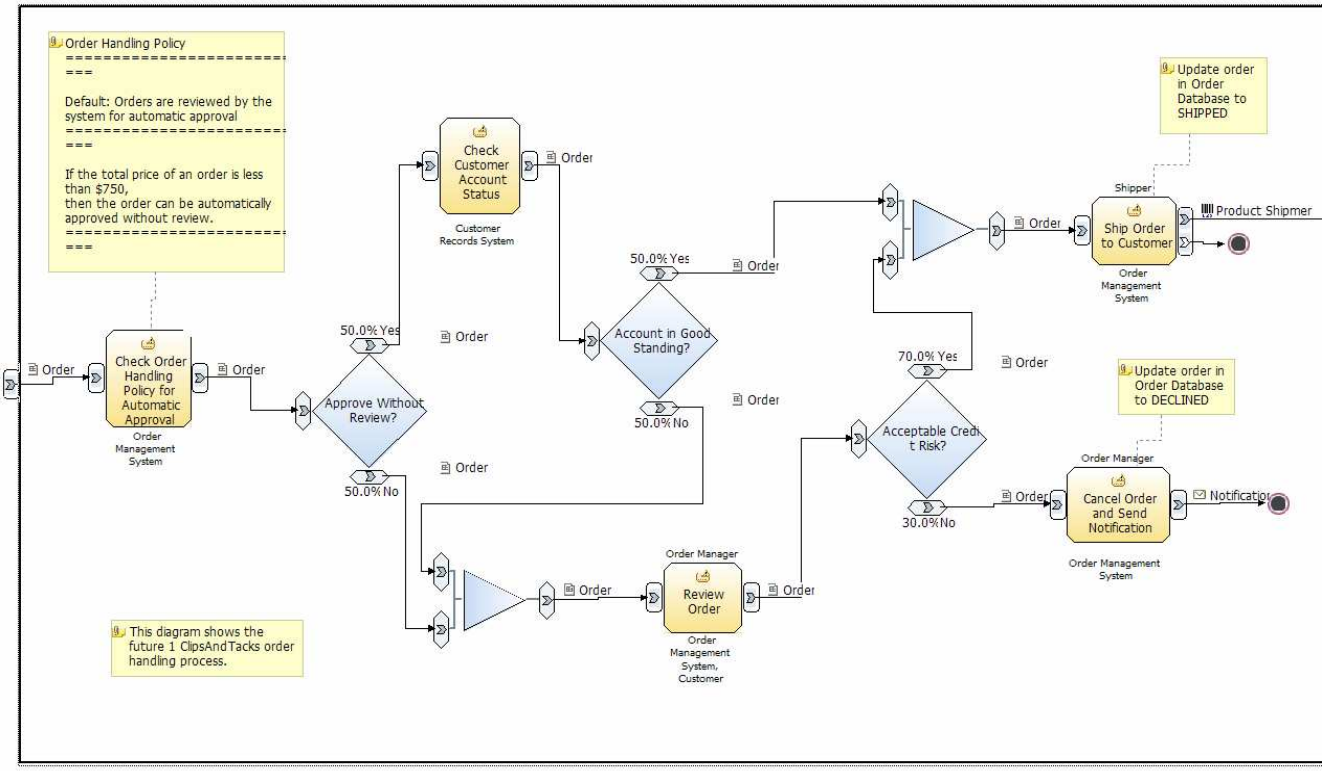
In this lab you will build a monitor model that has the same basic elements as the one that is defined in the Redbook. Namely you will create monitoring constructs in support of two KPIs:

- Average order fulfillment is 3 days or less
- Percentage of shipped orders is greater than 90%

You will also create situation events if declined orders are greater than or equal to 3 and if the order fulfillment time is greater than 3 days.

For dimensional analysis, you will create a Location dimension which allows you to drill down on country and city. You will also add an Order Status metric, which shows orders as ‘New’, ‘Cancelled’ or ‘Shipped’. And you will add measures for average order price, sum of order price for all orders, and order count.

Here is a diagram of the ClipsAndTacks process model:



This is the ordering process for the Clips And Tacks company. In this process, orders are received and optionally submitted to a review process. Orders which pass all checks are shipped to the customer. Other orders may be cancelled.

In the monitor model, you will create one monitoring context for each customer order. You will identify the inbound event that represents the receipt of the order, and this is marked so that it will create a new monitoring context instance for the order. You will also identify the inbound event that represents shipping the order, and this is marked so that it will trigger the termination of the monitoring context instance for the order. Since an order might also be cancelled, you will identify the inbound event that represents cancellation of the order, and this is also marked so that it will trigger the termination of the monitoring context instance.

URL Cheatsheet

The following URLs may be helpful to you as you exercise this lab. Note that the port numbers in the URL of your installation may be different depending on your configuration.

- Server administrative console

  http://localhost:9061/ibm/console/

- Web dashboard

  http://localhost:9081/BusinessDashboard/

Part 1: Create the monitor model

To create the monitor model, you will use the monitor model editor in Rational Application Developer or WebSphere Integration Developer. Note that the screen captures in this lab are based on WebSphere Integration Developer, so they may differ slightly from yours if you are using Rational Application Developer.

Note that there is a 'Problems' tab that displays errors concerning the model. Periodically, you should check this view to see if you have any problems that need to be addressed. Normally, warnings and informational messages are not a problem, but errors should be addressed.

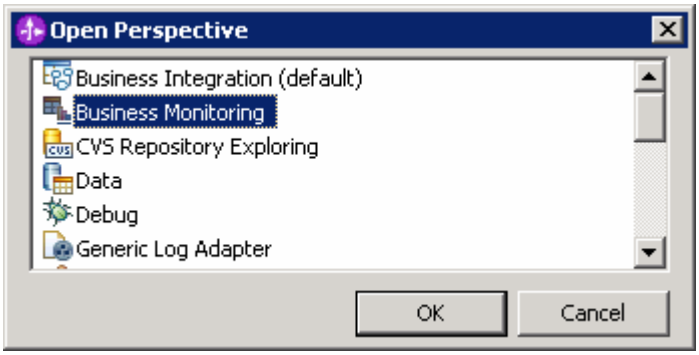
If you want to skip this section, then a solution has been provided. You can import the supplied monitor model into Rational Application Developer or WebSphere Integration Developer, and then proceed to the next section. Refer to Appendix 1, and then proceed to the next section.

____ 1. Start Rational Application Developer or WebSphere Integration Developer and setup the environment.

__ a. Start Rational Application Developer or WebSphere Integration Developer, and when prompted point to a new workspace such as C:\workspaces\ClipsAndTacksXSD

__ b. Close the Welcome tab

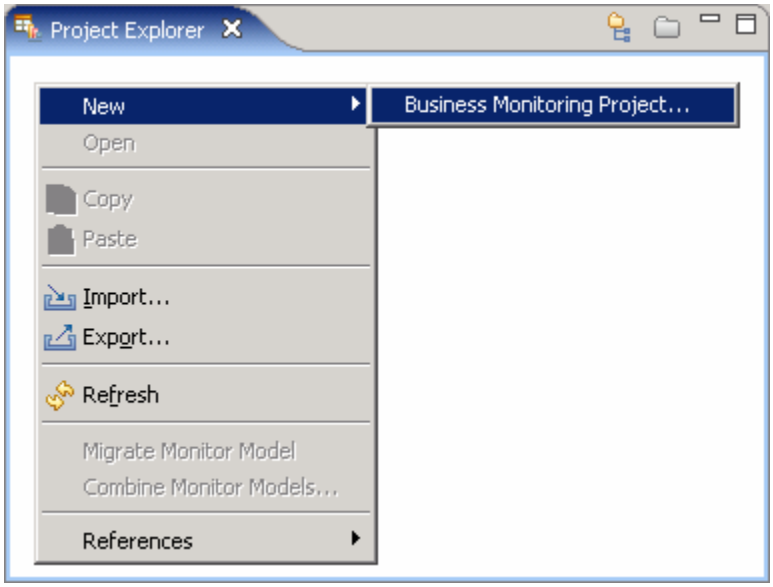
__ c. By default, you are in the Business Integration perspective. But you need to open the Business Monitoring perspective. From the main menu, select **Window → Open Perspective → Other**. The 'Open Perspective' dialog opens



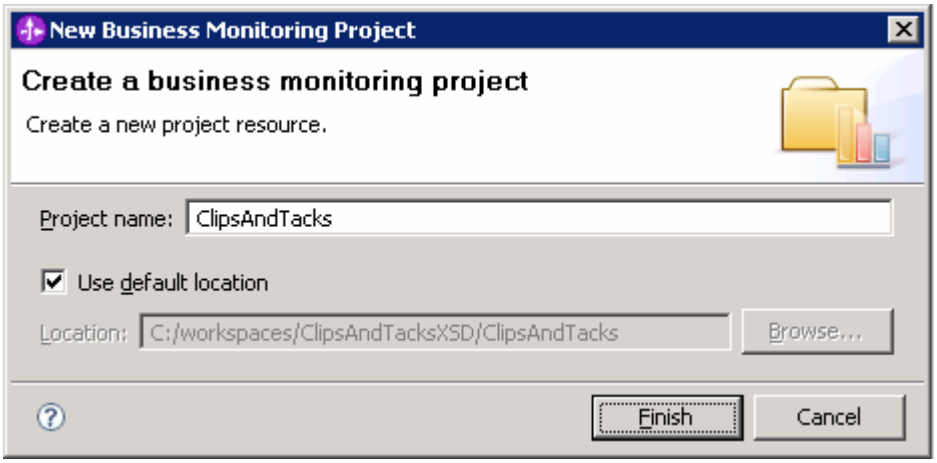
__ d. Select '**Business Monitoring**' from the 'Open Perspective' dialog and click **OK**

____ 2. Create a new monitoring project. A project is a container for your monitor models and event definitions.

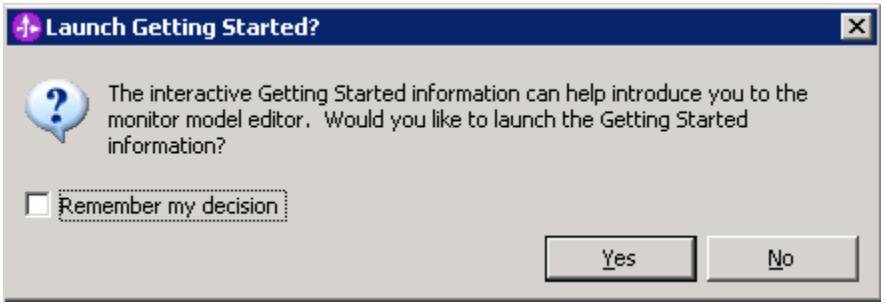
__ a. Right-click inside **Business Monitoring** project explorer (top left view in the Business Monitoring Perspective) and select **New → Business Monitoring Project…** from the pop-up menu. The 'New Business Monitoring Project' panel opens



__ b. In the 'New Business Monitoring Project', enter the 'Project name:' as **ClipsAndTacks**

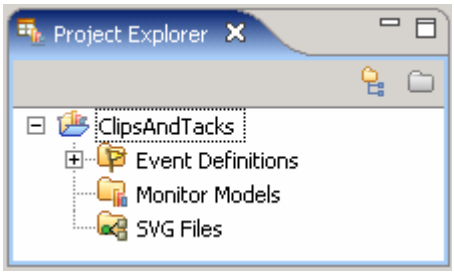


__ c. Click the **Finish** button. A question dialog pops up asking if you want to launch the Getting Started information



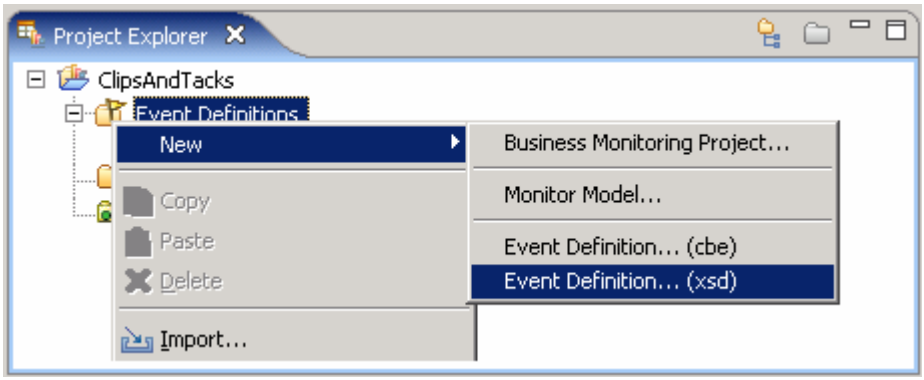
__ d. Click the **Yes** button to if you want to launch the Getting Started information

__ e. A new project named **ClipsAndTacks** is created as shown below:

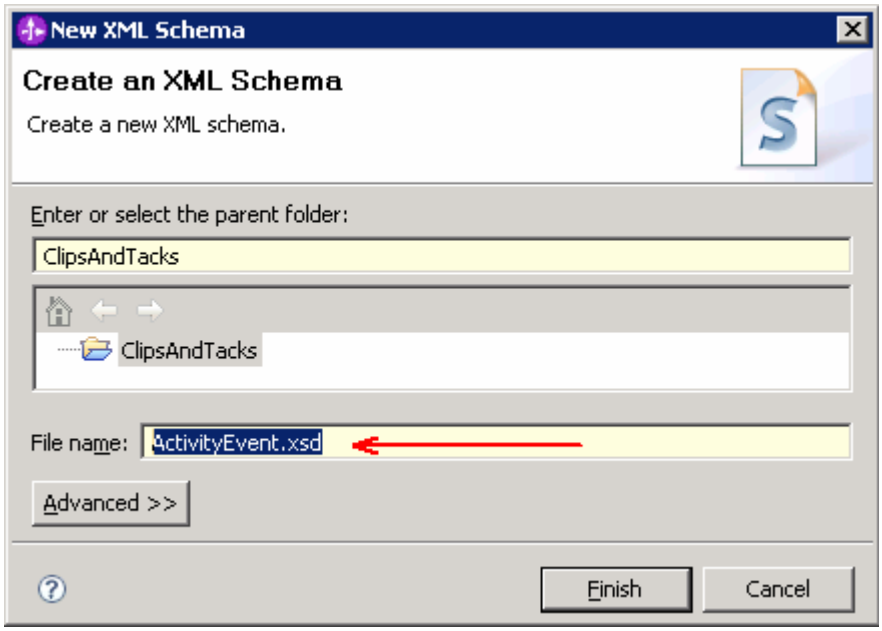


____ 3. Create a new XSD event definition named, **ActivityEvent**

__ a. In the Business Monitoring project explorer view, expand **ClipsAndTacks**, right-click on **Event Definitions** and select **New → Event Definition… (xsd)** from the pop-up menu

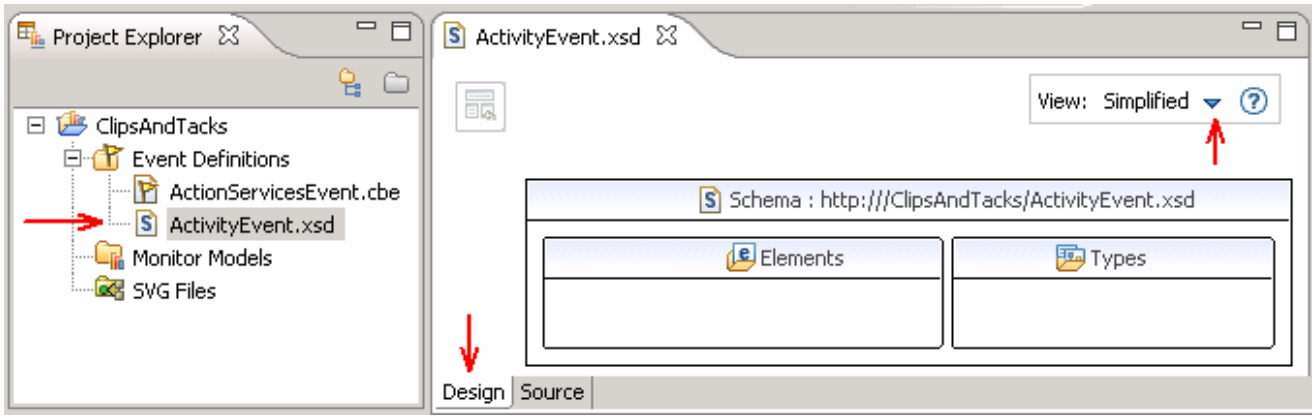


__ b. The '**New XSD Schema**' panel opens as shown below:

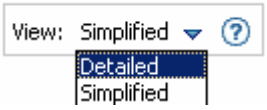


__ c. Enter '**ActivityEvent.xsd**' as 'File name:' and click **Finish**

__ d. The schema you created is listed under the event definitions category and the schema is opened in a simplified XSD editor view as shown below:

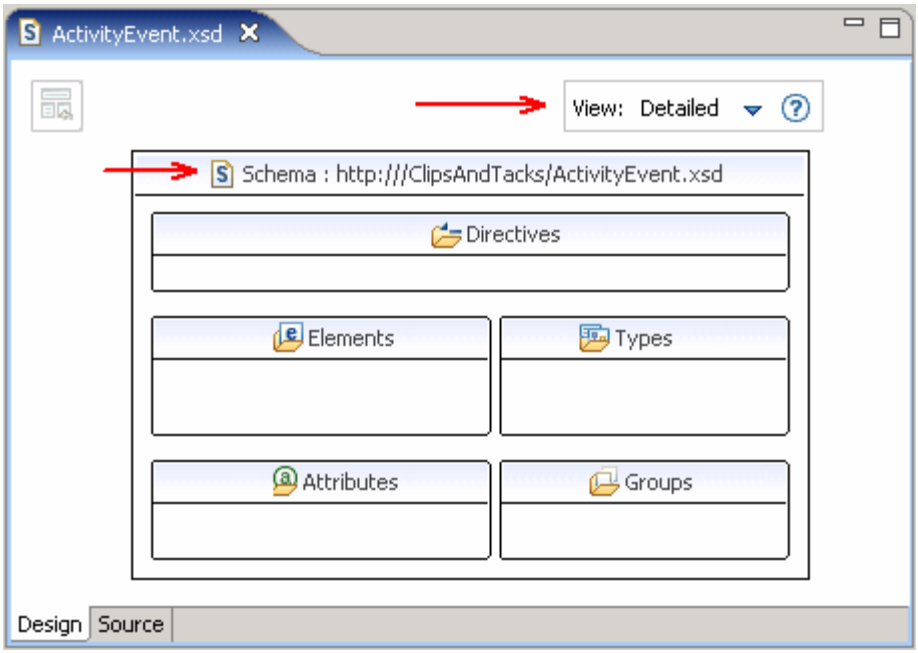


__ e. You can use two views to edit an XML schema file, 'Simplified' or 'Detailed' view. Use the Simplified view if you are new to XML schemas. Use the Detailed view if you are familiar with XML schemas and their more complicated constructs (for example, groups, choices, attribute references, and directives). To change the XSD editor view from '**Simplified**' to '**Detailed**' click on the pull down menu icon (▾) located on the top right corner



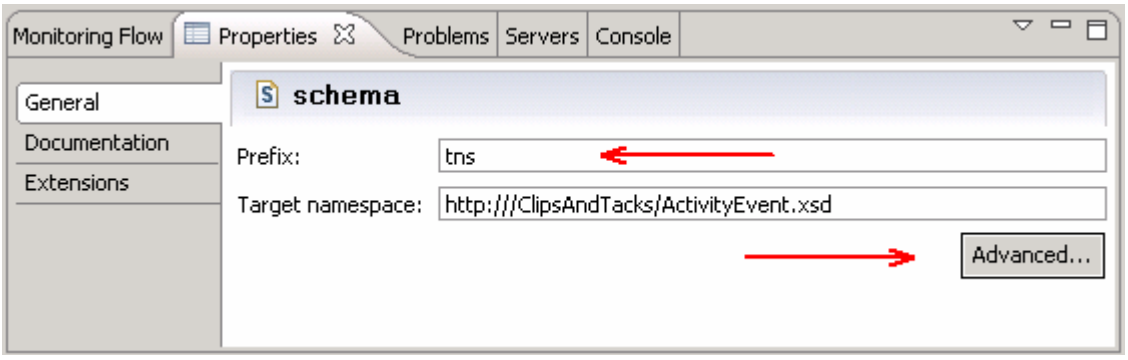
__ f. Click **OK** to confirm switching the XSD editor from simplified to detailed view mode

__ g. The detailed view mode of the XSD editor is as shown below:



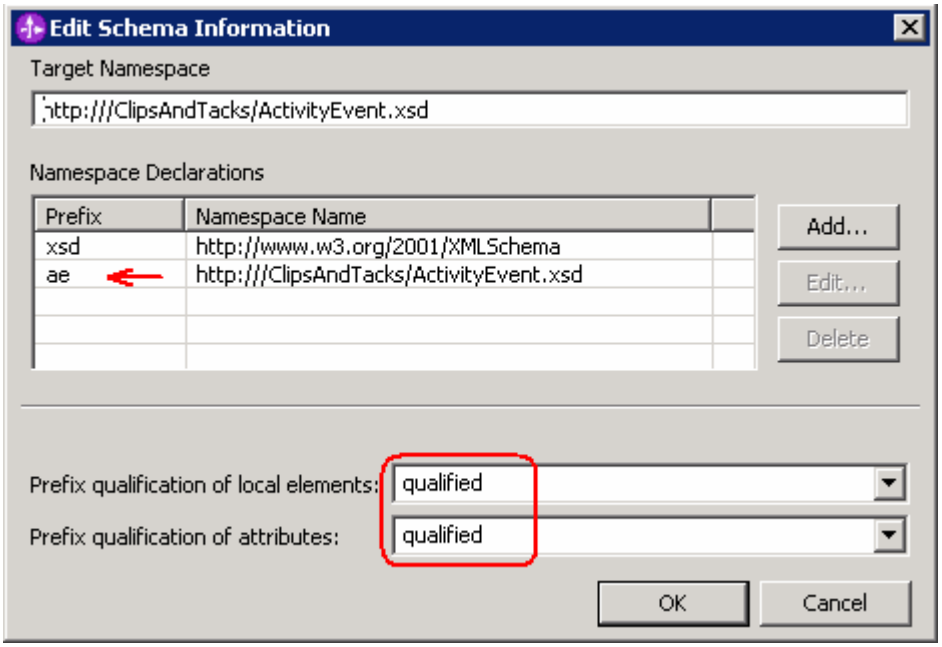
____ 4. Review and update the <schema> element, which is the root element of the schema

__ a. In the XSD editor, select the schema element and then select the '**Details**' tab of the '**Properties**' view



__ b. Update the '**Prefix:**' from **tns** to **ae** (where **ae** stands for activity event and the prefix resembles the name of the schema)

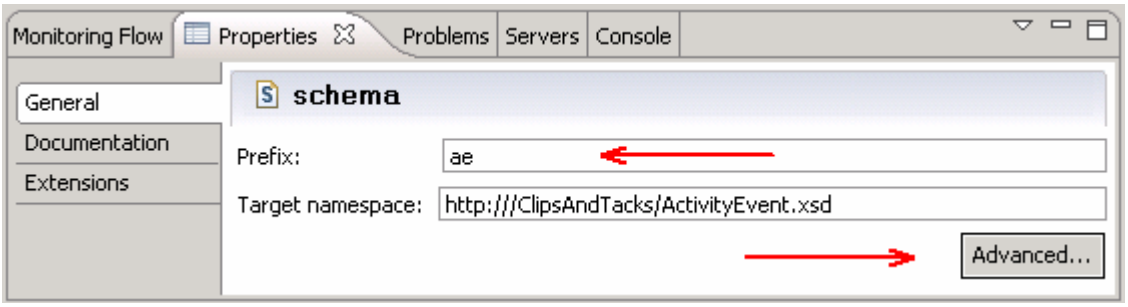
__ c. Click the **Advanced** button to edit the schema information (attributes)



__ d. Ensure that the prefix qualification of local elements and the attributes is set to '**qualified**'

Note: Prefix qualification indicates whether locally declared elements or attributes must be qualified by the target namespace in an instance document. If the value of this attribute is 'unqualified', then locally declared elements should not be qualified by the target namespace. If the value of this attribute is 'qualified', then locally declared elements must be qualified by the target namespace. Using 'qualified' helps to ensure that expressions written to access pieces of information within a type are unambiguous.

__ e. Click **OK**. The '**General**' properties view of the 'ActivityEvent' schema should look like the picture below:



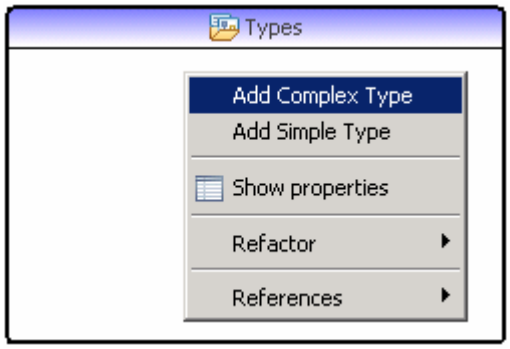
__ f. Save the changes. **File → Save** or **Ctrl + S**

____ 5. Add a complex element type named, **ActivityEventData** and this sequence of elements to the complex type

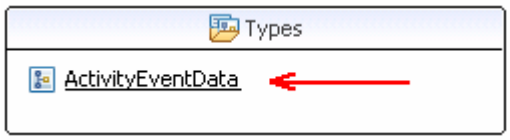
These are the elements - including the data types - that you are adding:

Name of the Element	Type
businessUnit	string
processName	string
mcID	string
activityName	string
eventType	string
startTime	dateTime
endTime	dateTime

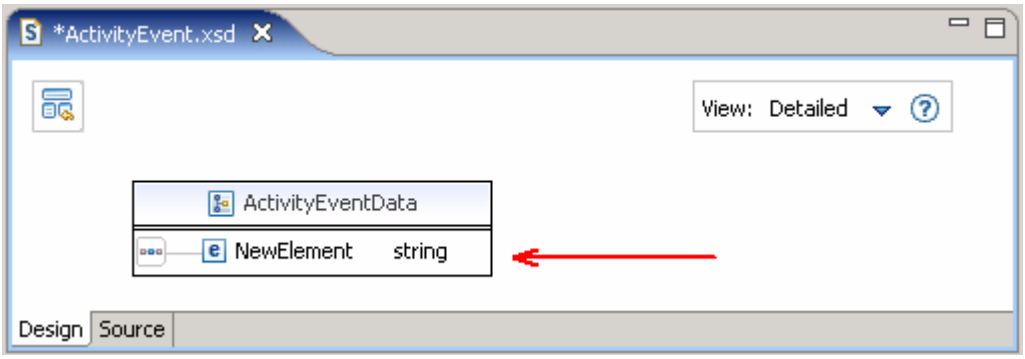
__ a. Right click in the **Types** section of the XSD editor and select '**Add Complex Type**' from the pop-up menu



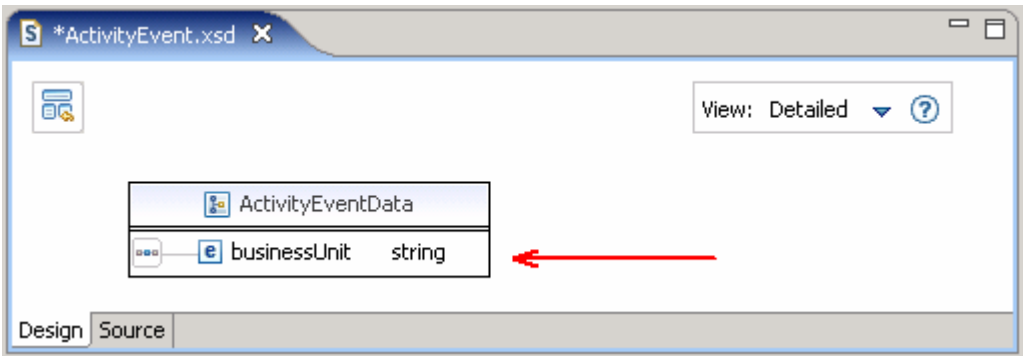
__ b. Name the complex type as **ActivityEventData** and save the changes. You should see the new complex type element added, as shown below:



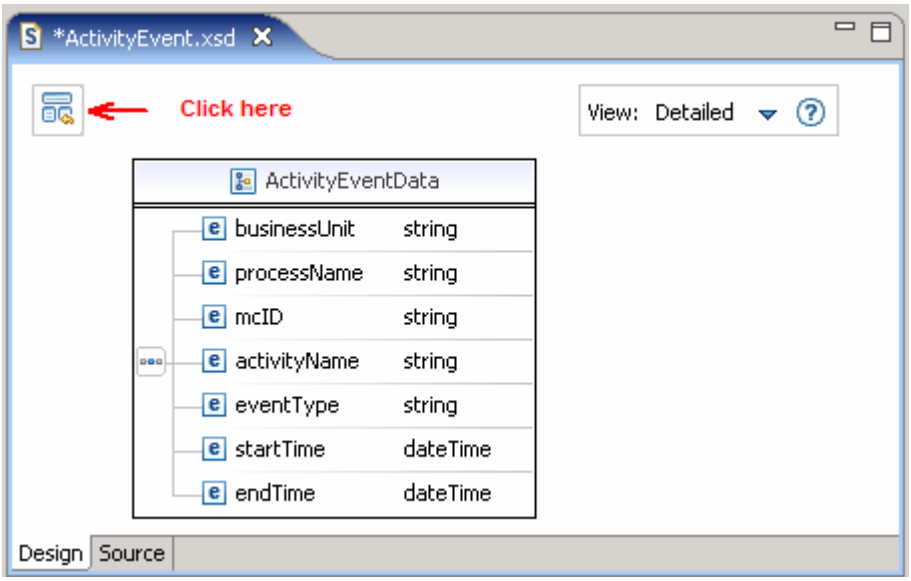
__ c. Right click on the complex type '**ActivityEventData**' again and select '**Add Element**' from the pop-up menu. By default the new element is created with a default name '**NewElement**' and xsd type '**string**' as shown below:



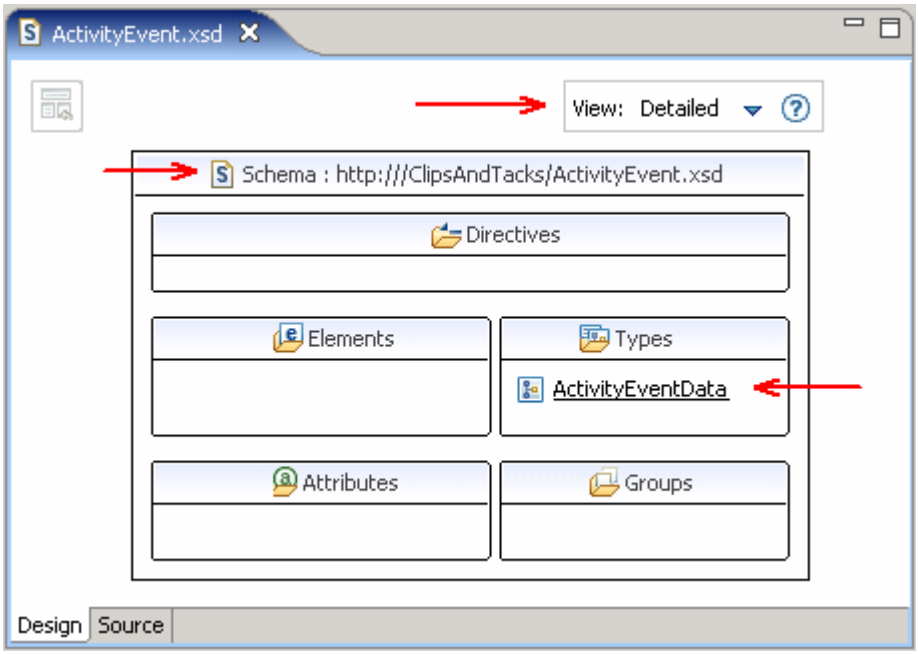
__ d. Modify the element name from 'NewElement' to '**businessUnit**' and accept the default xsd type as '**string**'



__ e. Repeat the above instructions to add the remaining elements mentioned in the table. The final '**ActivityEventData**' complex type should look like the picture below:



__ f. Now click the icon ( ) located on the left corner. This leads you back to the full schema editor view as shown below:



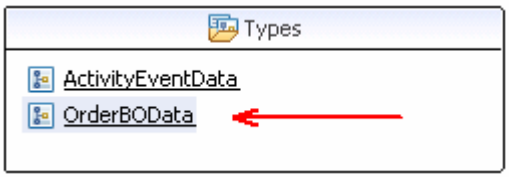
__ g. Save the changes. **File → Save** or **Ctrl + S**

____ 6. Add another complex element type named **OrderBOData** and this sequence of elements to the complex type

These are the elements - including the data types - that you are adding:

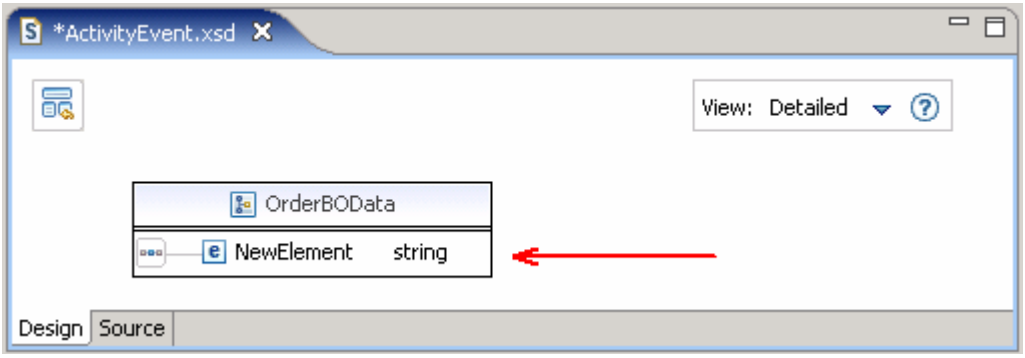
Name of the Element	Type
orderNumber	string
customerNumber	string
orderState	string
City	string
Country	string
productNumber	string
Quantity	int
totalPrice	float

__ a. Right click in the **Types** section of the XSD editor and select '**Add Complex Type**' from the pop-up menu

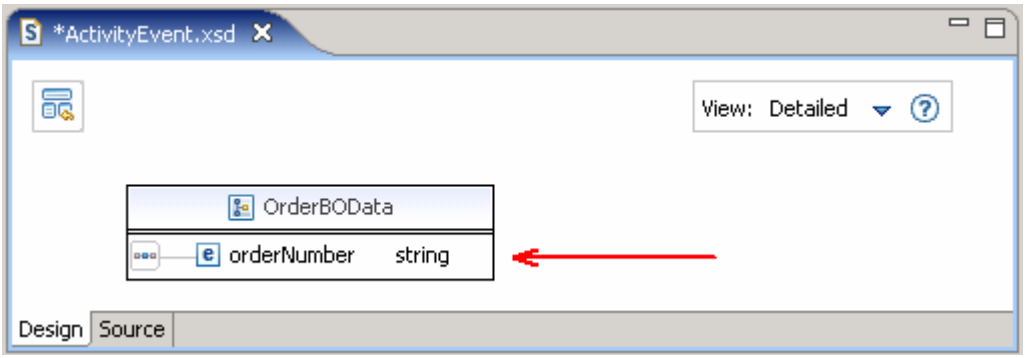


__ b. Now right click on the complex type '**OrderBOData**' and select '**Add Sequence**' from the pop-up menu. Note that a sequence is automatically added when you add an element to the complex type.

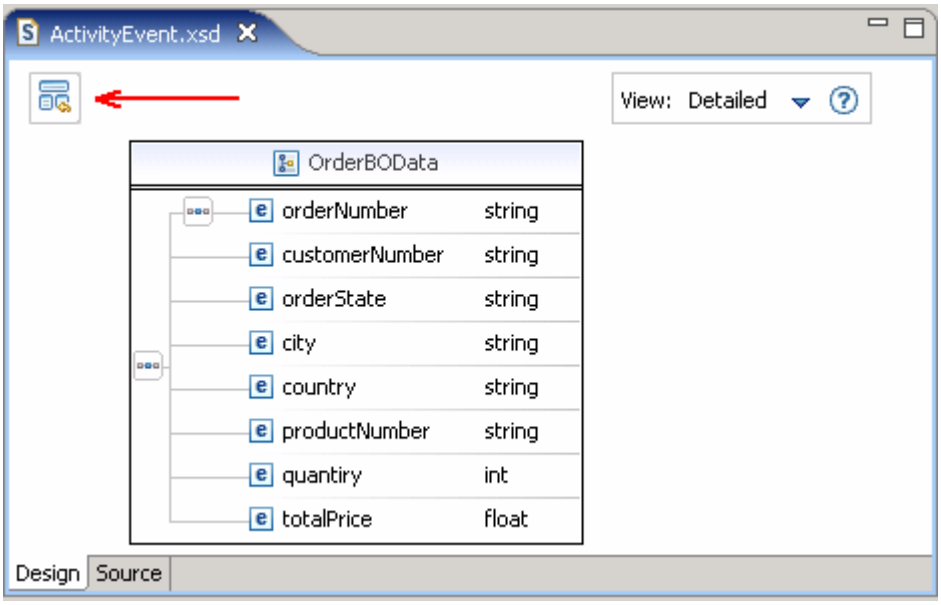
__ c. Right click on the complex type '**OrderBOData**' again and select '**Add Element**' from the pop-up menu. By default the new element is created with a default name '**NewElement**' and xsd type '**string**' as shown below:



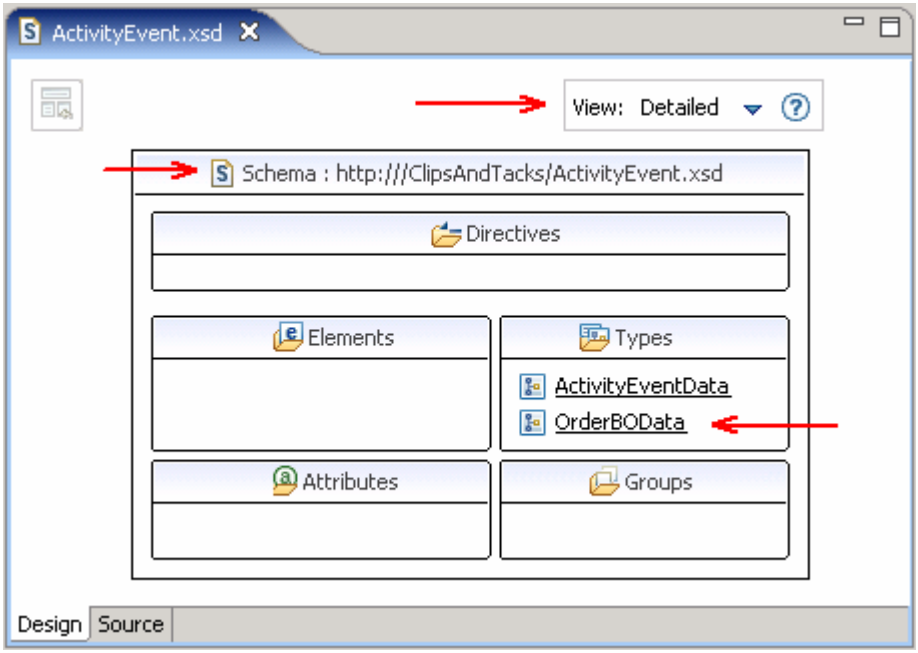
__ d. Modify the element name form 'NewElement' to '**orderNumber**' and accept the default xsd type as '**string**'



__ e. Repeat the above instructions to add the remaining elements mentioned in the table. The final '**OrderBOData**' complex type should look like the picture below:

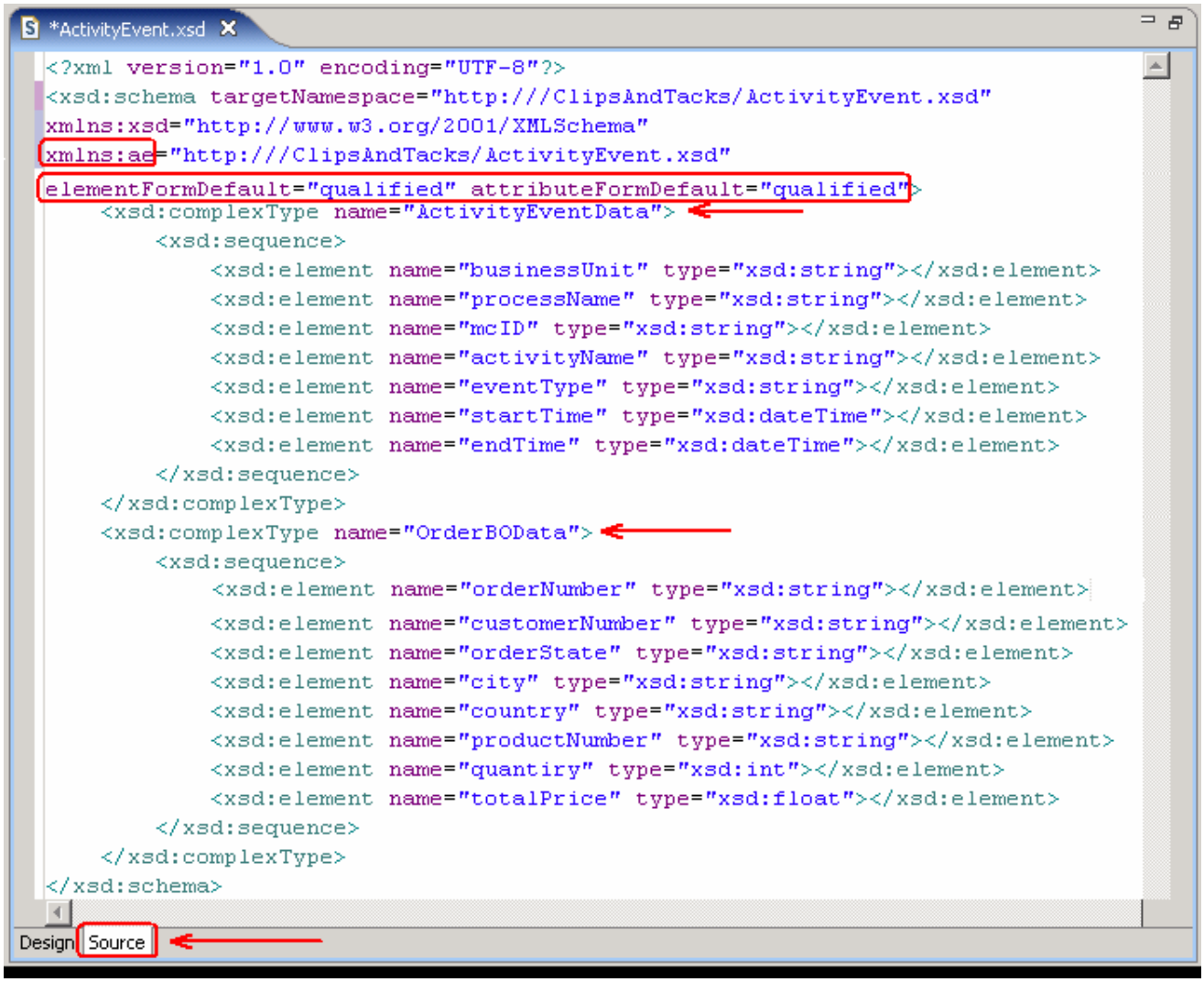


__ f. Now click the icon ( ) located on the left corner. This leads you back to the full schema editor view as shown below:



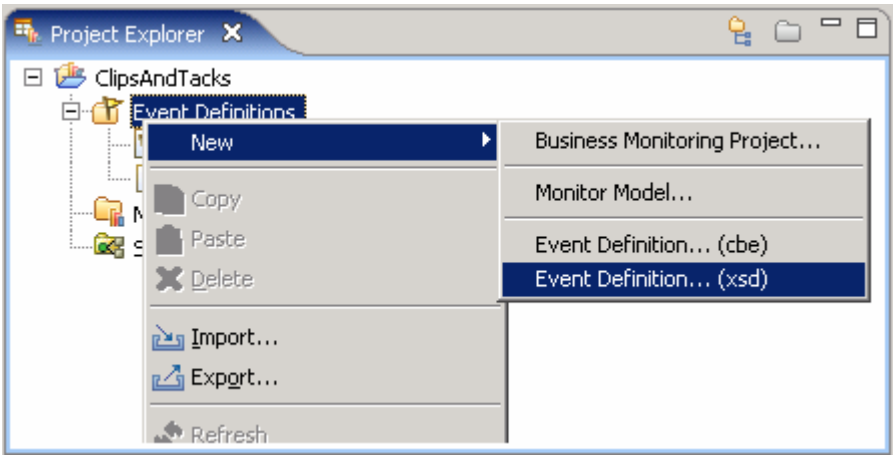
__ g. Save the changes. **File → Save** or **Ctrl + S**

____ 7. The following is the source of the ActivityEvent schema you created:



____ 8. WebSphere Business Monitor 6.1 has support for concurrent versions of monitor models. In order to take advantage of the Monitor server's version handling capabilities, each monitoring context should have a unique event definition that is the single inbound event that has the setting 'If no instances are found: Create new instance'. One can think of this as one inbound event carrying 'create monitor context' semantics, and all others as 'update monitor context' semantics. Clips And Tacks is a simple sample and could use a single event definition for the process. However, to satisfy the requirement of a unique event definition for version handling, you will create a new event definition that will create new instances of the monitor model.

__ a. In the project explorer, right click on **Event Definitions** and select **New → Event Definition… (xsd)** from the pop-up menu. The 'New' event definition panel opens

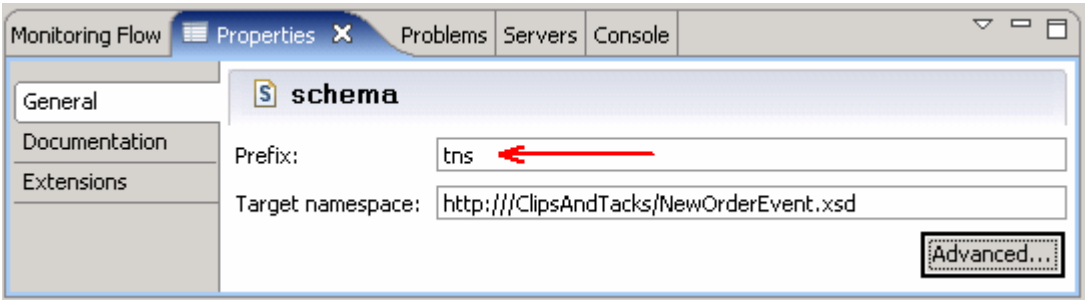


__ b. In the **New** event definition panel, select **ClipsAndTacks** as the parent folder and enter the 'File name:' as **NewOrderEvent.xsd** and click **Finish**

__ c. Click '**Yes**' over the question dialog if it appears

____ 9. Review and update the <schema> element, which is the root element of the schema

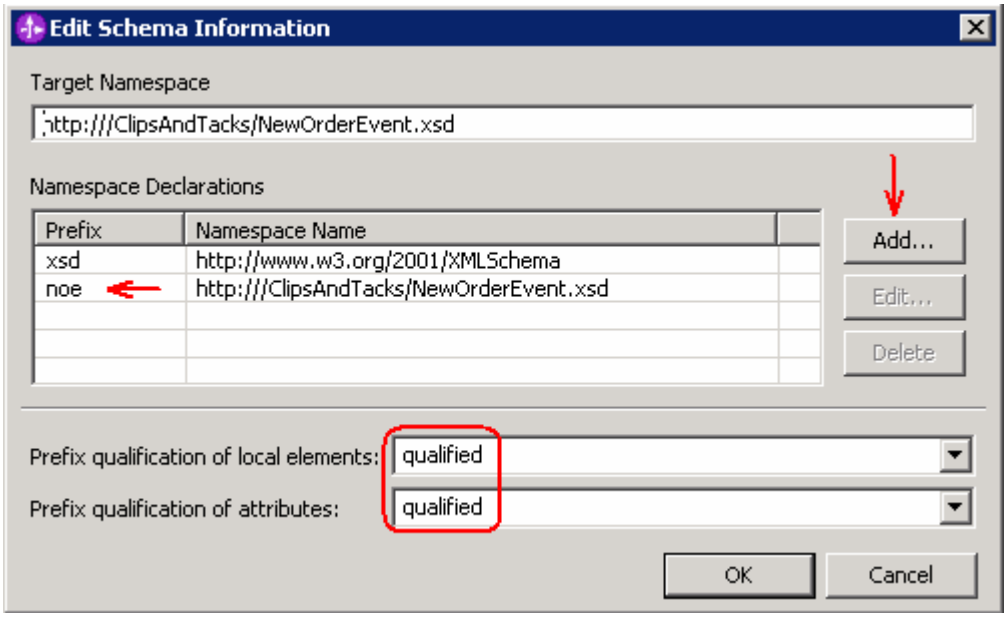
__ a. In the XSD editor, select the schema element and then select the '**General**' tab of the '**Properties**' view



__ b. Update the '**Prefix:**' from **tns** to **noe**

Note: The '**Target NameSpace Prefix**' namespace can be given of your choice. 'noe' is short for 'NewOrderEvent'

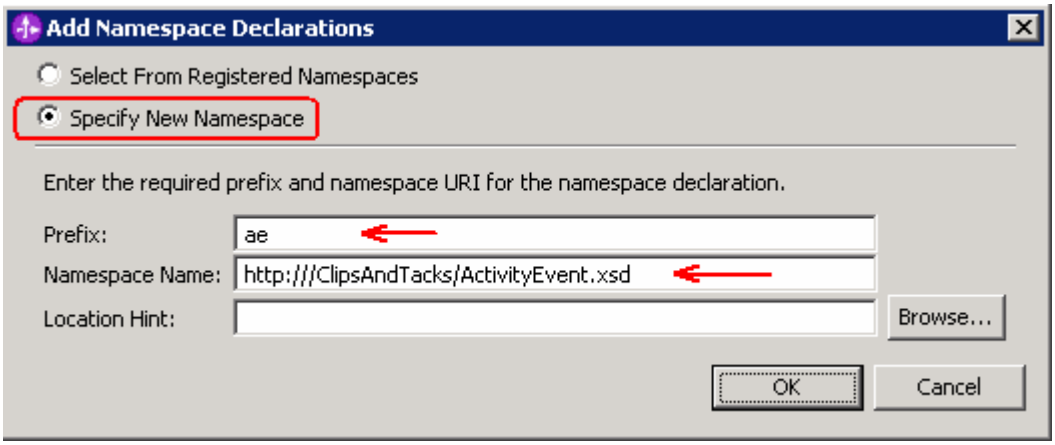
__ c. Click the **Advanced** button to edit the schema information (attributes)



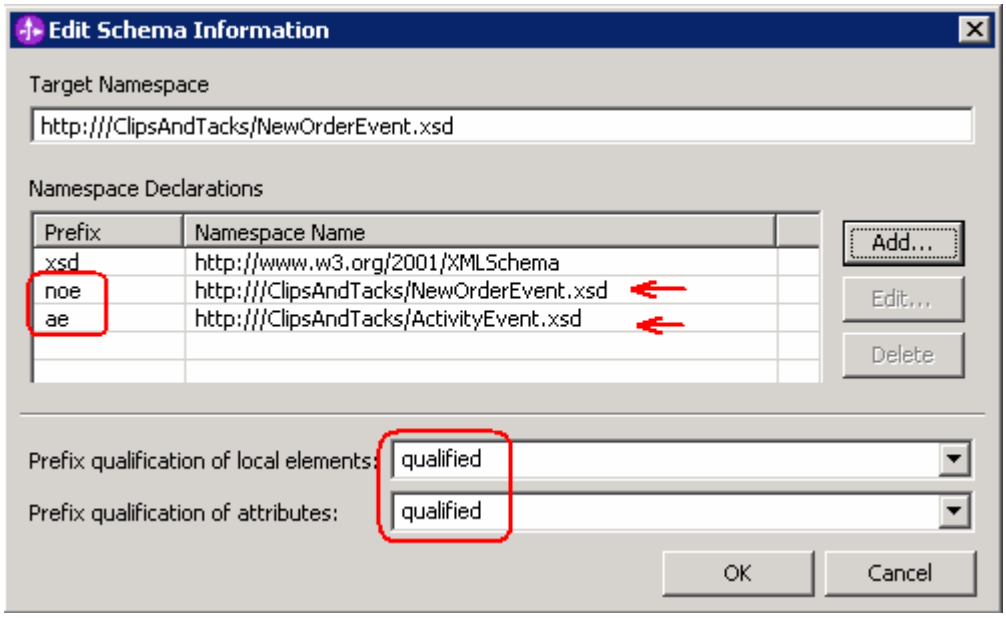
__ d. Click the **Add** button to add the **ActivityEvent** namespace so the 'ActivityEvent' schema can be imported to the current 'NewOrderEvent' schema. The '**Add Namespace Declarations**' panel opens

__ e. Enter these parameters:

- Select the radio button for '**Specify New Namespace**'
- Prefix : **ae**
- Namespace Name : **http:///ClipsAndTacks/ActivityEvent.xsd**

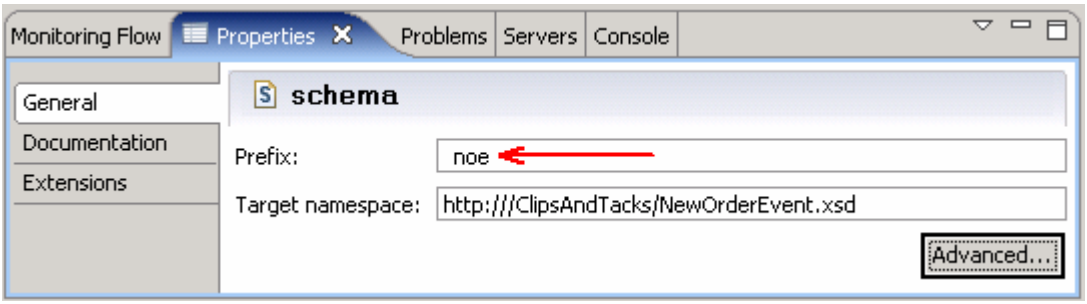


__ f. Click **OK**



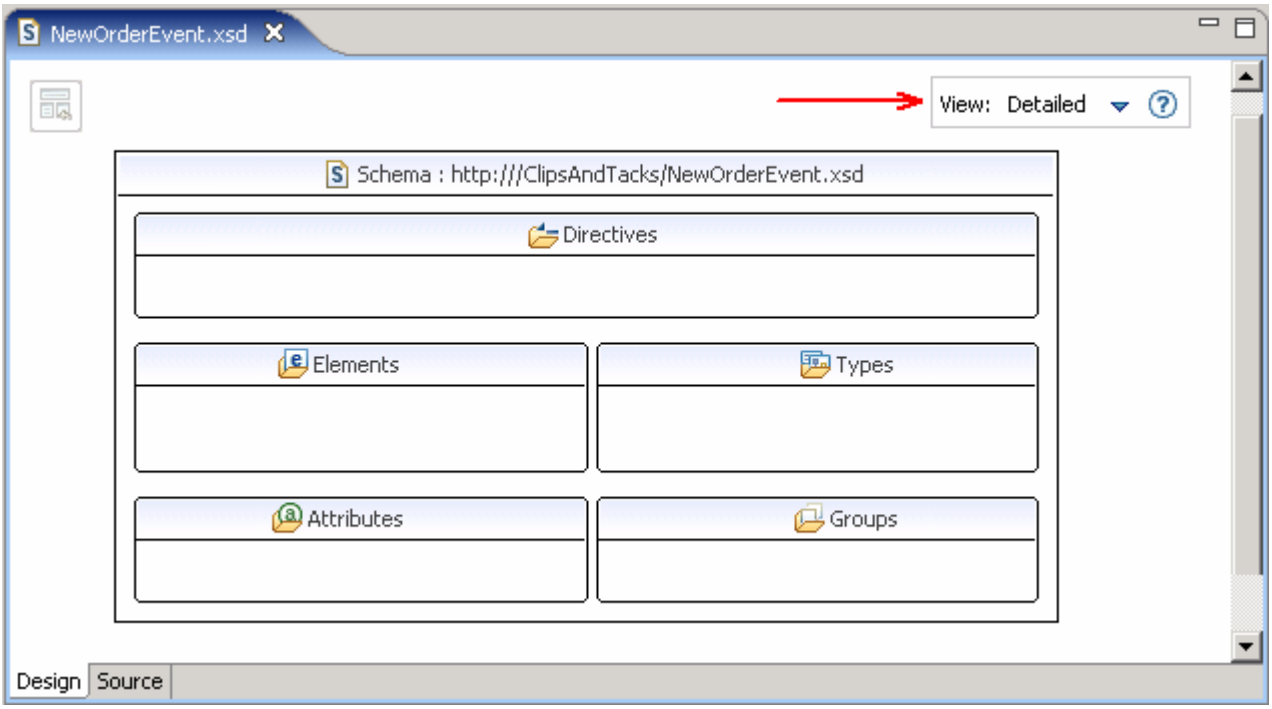
__ g. Ensure that the prefix qualification of local elements and the attributes is set to '**qualified**'

__ h. Click **OK**. The '**General**' properties view of the 'ActivityEvent' schema should look like the picture below:

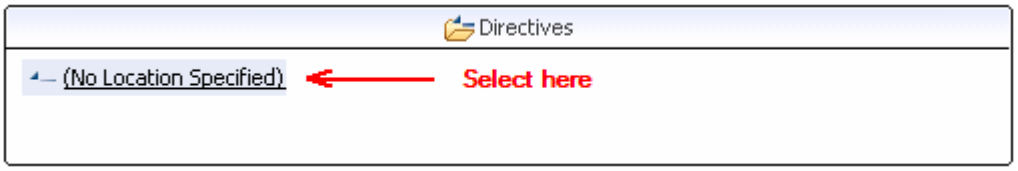


__ i. Save the changes. **File → Save** or **Ctrl + S**

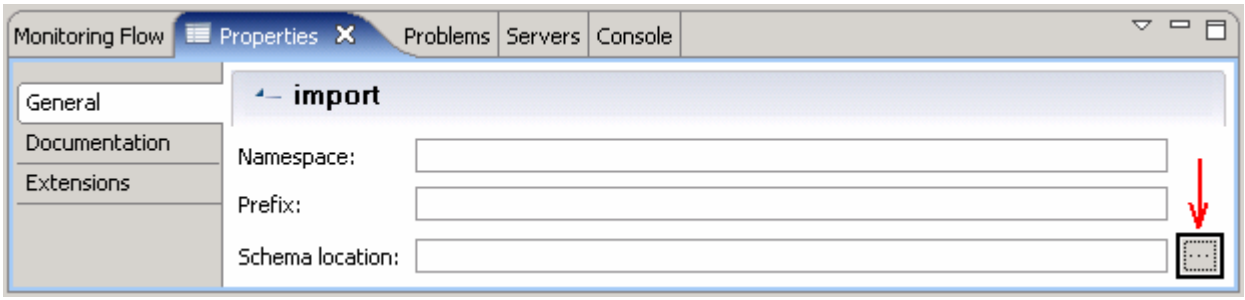
__ j. If the 'NewOrderEvent.xsd' schema opens in a '**Simplified**' XSD schema editor. Click on the pull-down button (▾) located on the top right corner of the XSD schema editor and select '**Detailed**' from the drop down list.



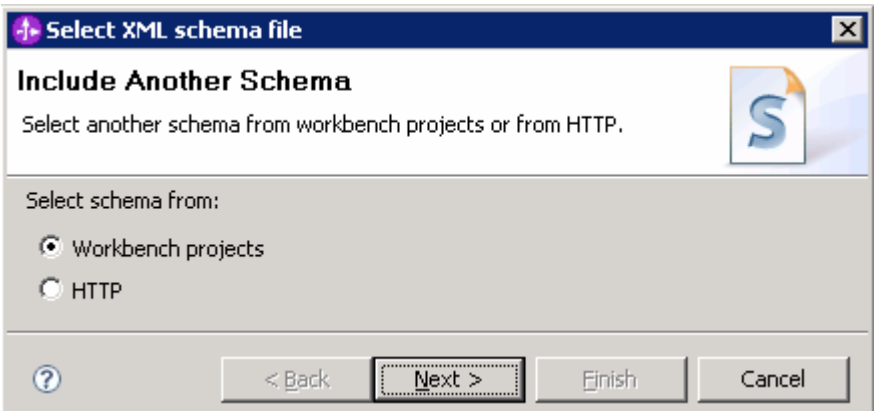
__ k. Right click any where in the '**Directives**' section and select '**Add Import**' from the pop-up menu to import the 'ActivityEvent' schema. An empty XSD <import> element is added



__ l. In the XSD editor, select the empty XSD <import> element and then select the 'Properties' General panel

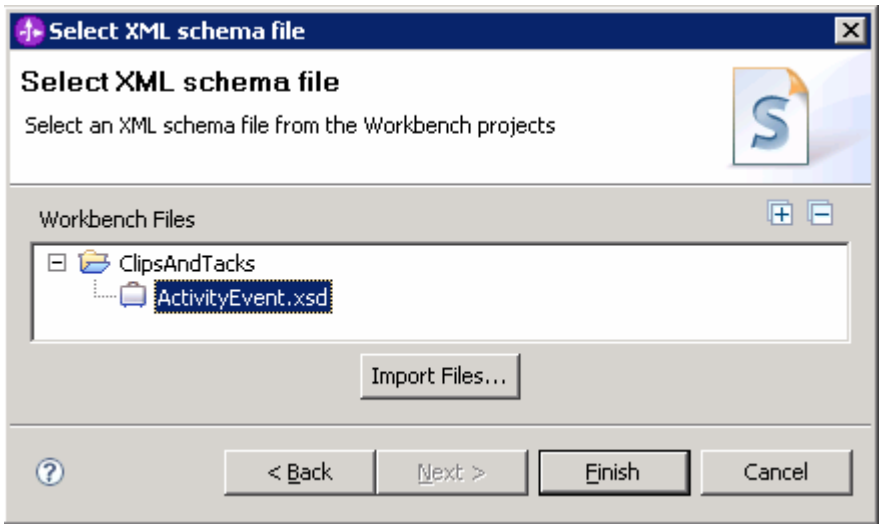


1) Click the button for '**Schema Location**'. The '**Select XML Schema file**' panel opens

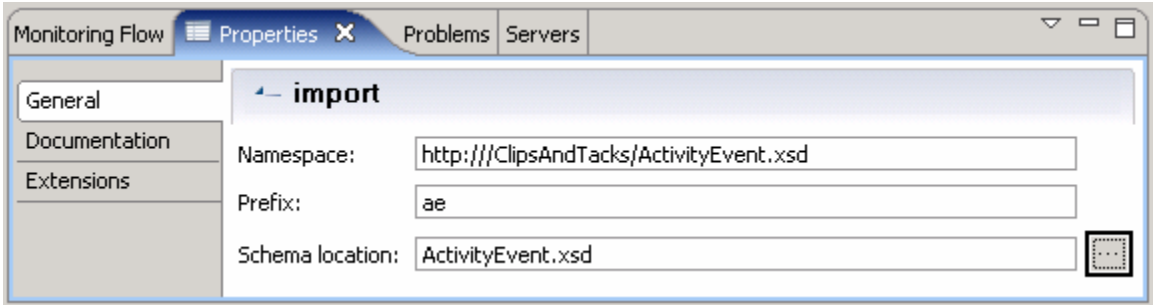


2) Ensure the radio button next to '**Workbench projects**' is selected and click **Next**

3) In this panel, expand the 'ClipsAndTacks' folder and then select 'ActivityEvent.xsd'

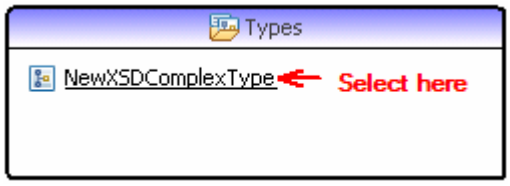


4) Click **Finish**. The Properties **General** panel should look the picture below:



5) Save the configuration. **File → Save** or **Ctrl + S**

__ m. Now right click in the '**Types**' section and select the '**Add Complex Type**' from the pop-up menu. An XSD <complexType> element is added with a default name



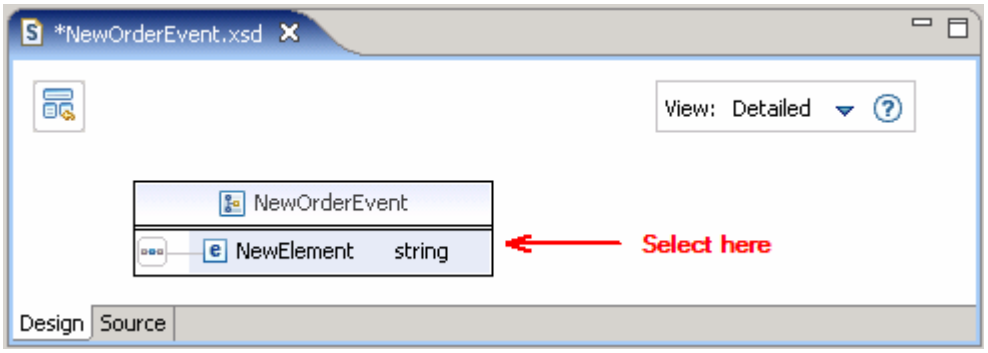
__ n. In the XSD schema editor, select the empty XSD <complexType> element and then select the 'Properties' General panel.

1) Rename the name of the complexType element to '**NewOrderEvent**'

2) Save the configuration. **File → Save** or **Ctrl + S**

__ o. Now right click on the '**NewOrderEvent**' complex type element and select '**Add Sequence**' from the pop-up menu

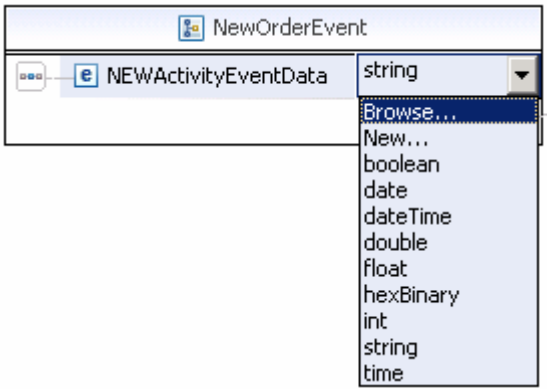
__ p. Right click on the '**NewOrderEvent**' complex type element again and select '**Add Element**' from the pop-up menu. A new element of type string is added with a default name as shown below:

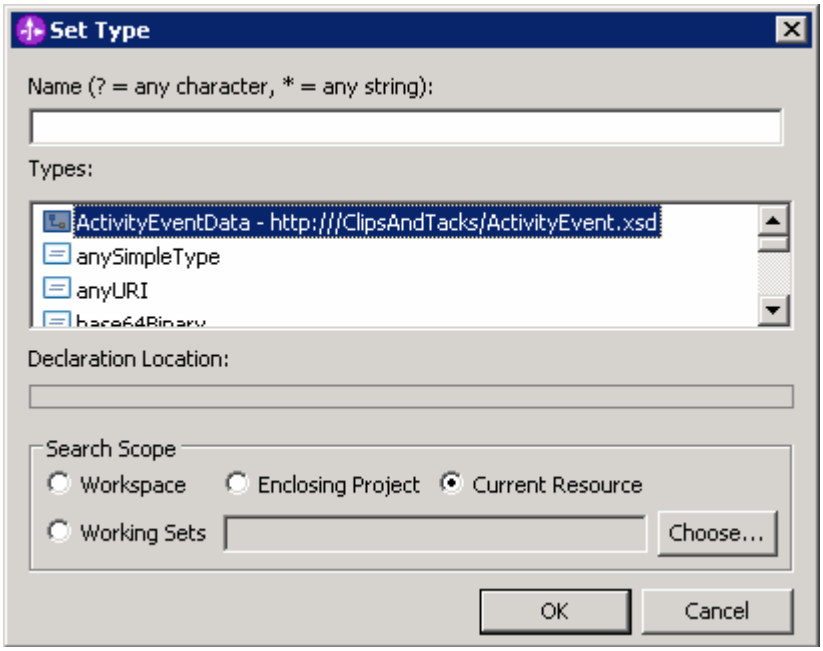


__ q. In the XSD schema editor, select the XSD <element> and then select the 'Properties' General panel

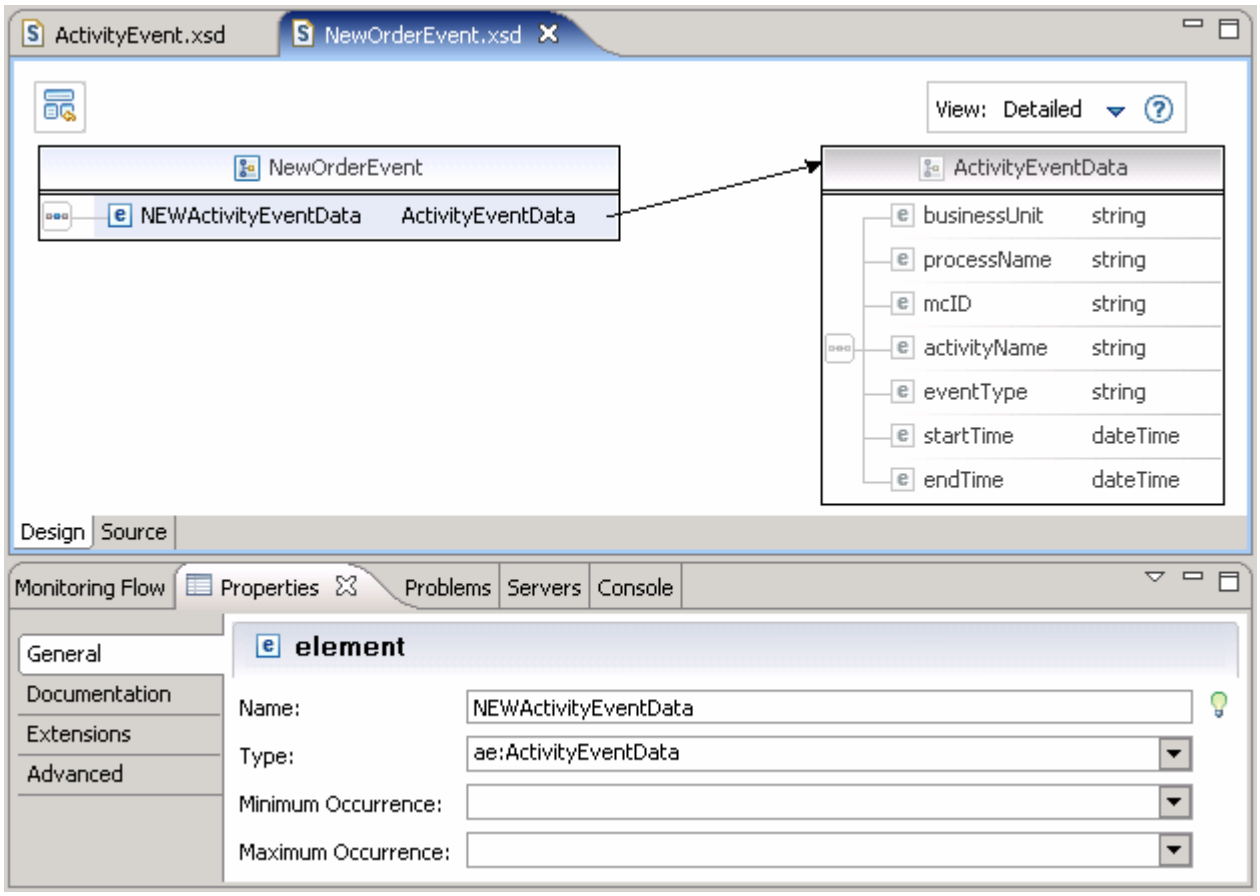
1) Rename the name of the element to '**NEWActivityEventData**'

2) For the '**Type:**' select '**Browse**' from the drop down list. The '**Set Type**' panel opens



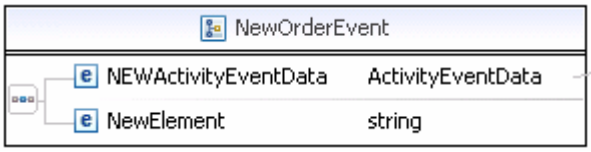


3) Select 'ActivityEventData' from the 'Types' text area and click **OK**. The XSD schema diagram and the properties view of the new '**NEWActivityEventData**' element should look like the picture below:



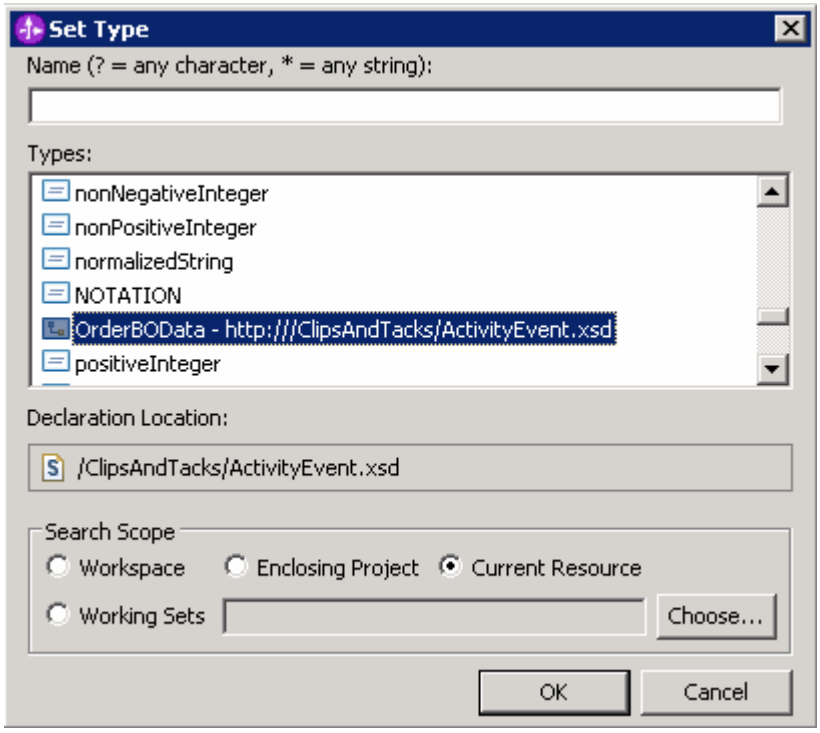
4) Save the configuration. **File → Save** or **Ctrl + S**

__ r. Now right click over the '**NewOrderEvent**' complex type element and select '**Add Element**' from the pop-up menu. A new element of type string is added with a default name as shown below:

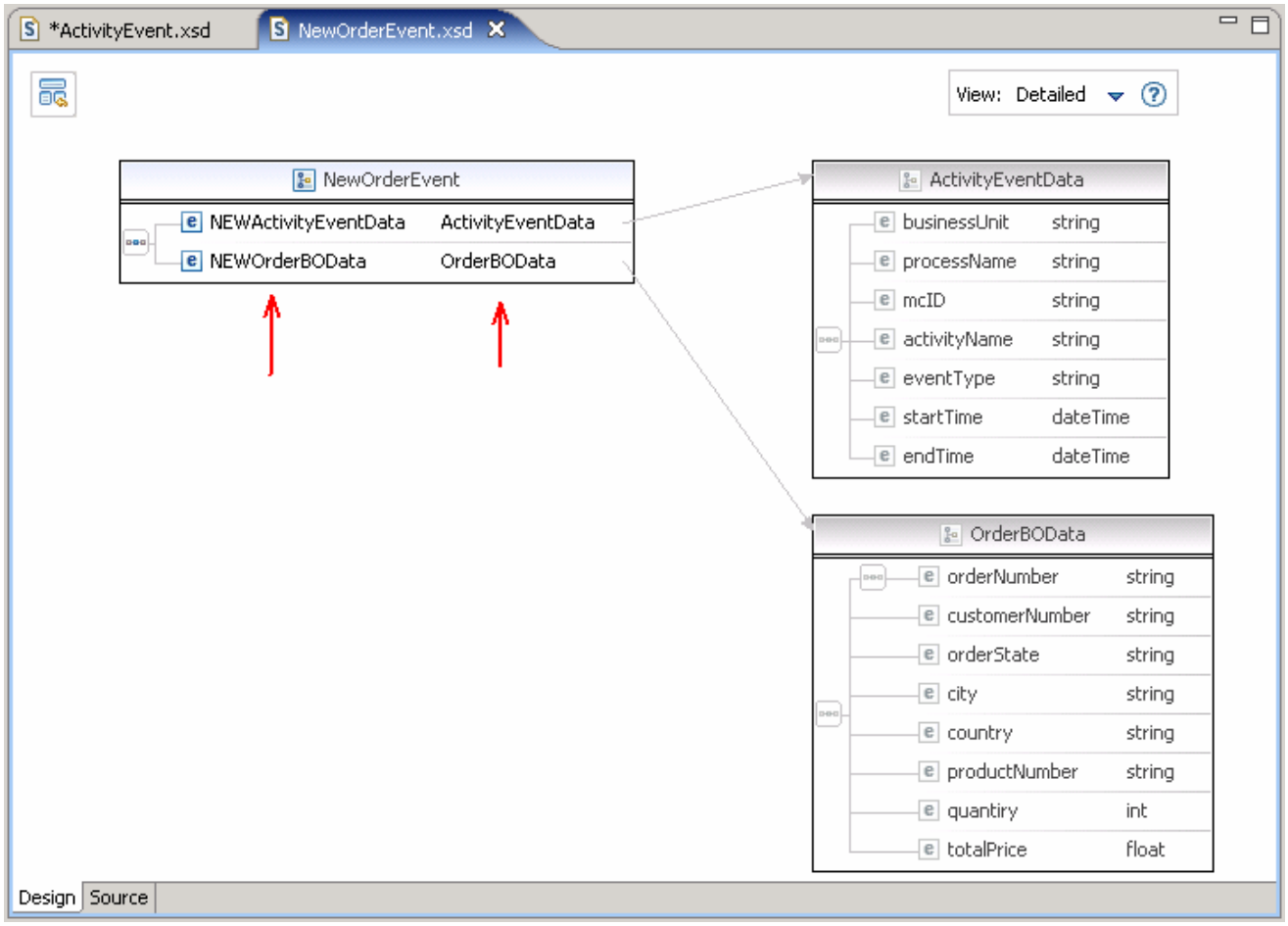


1) Rename the name of the element to '**NEWOrderBOData**'

2) For the '**Type:**' select '**Browse**' from the drop down list. The '**Set Type**' panel opens

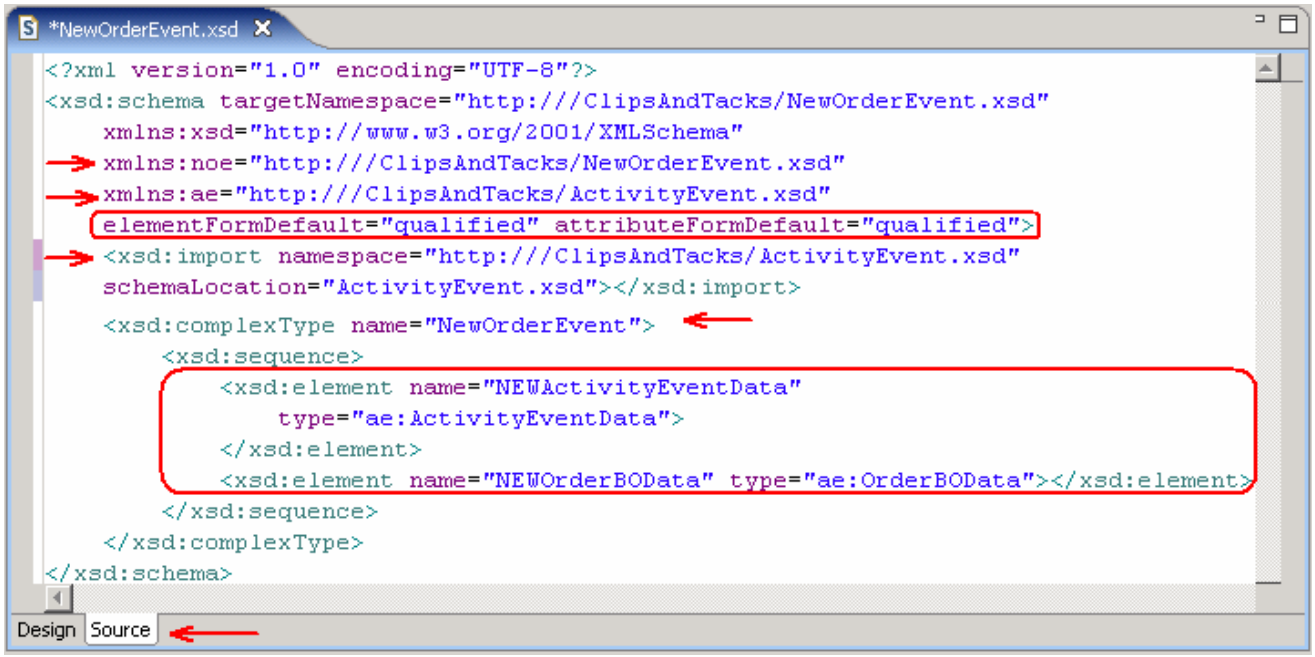


3) Select 'OrderBOData' from the 'Types' text area and click **OK**. The XSD schema diagram of '**NEWOrderBOData**' element should look like the picture below:



4) Save the configuration. **File → Save** or **Ctrl + S**

__ s. The following is the source of the NewOrderEvent schema you created:

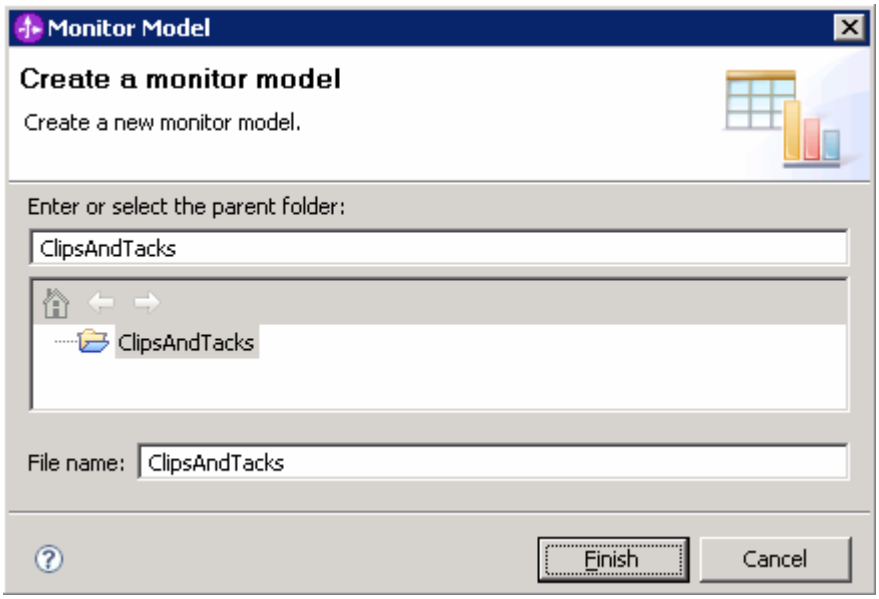


__ t. Close the XSD schema editor

____ 10. Create a new monitor model ClipsAndTacks. The monitor model contains the metrics and KPIs that you want to monitor and it is the source for creating the deployment code for the monitor server.

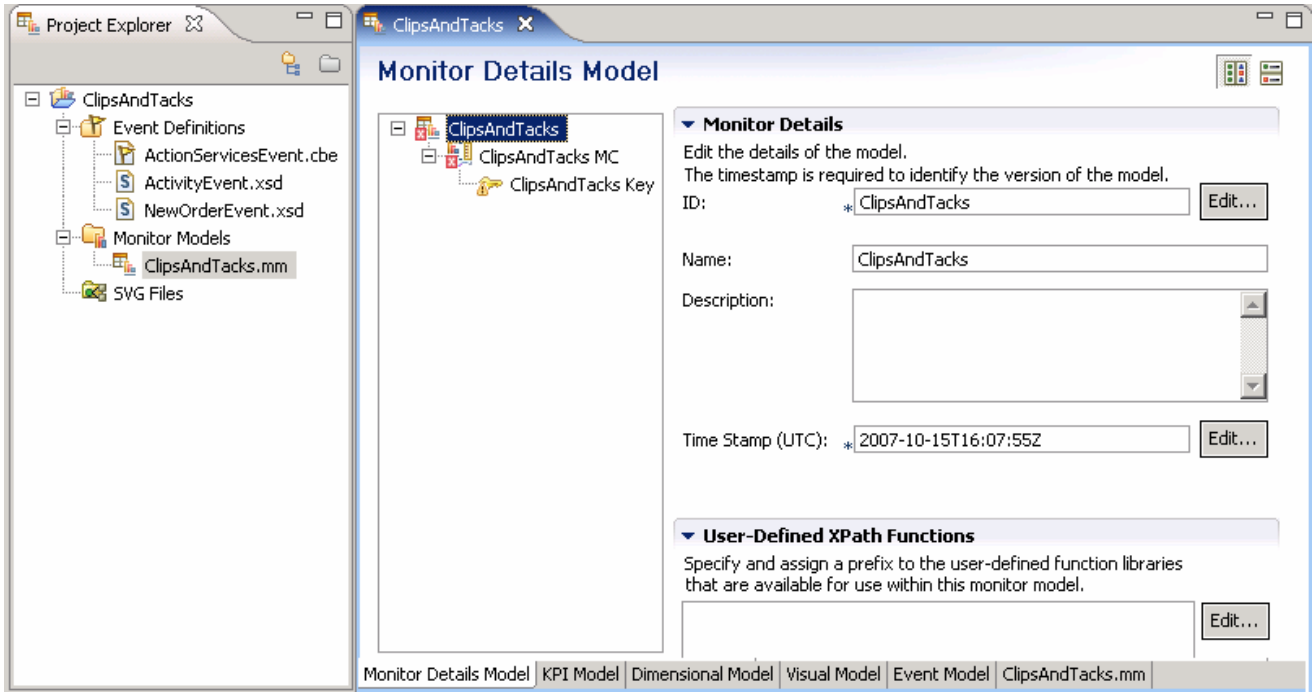
__ a. Right click in the 'Business Monitoring' project explorer view and select **New → Monitor Model…** from the pop-up menu. The Monitor Model panel opens

1) For the 'File name:', enter **ClipsAndTacks**



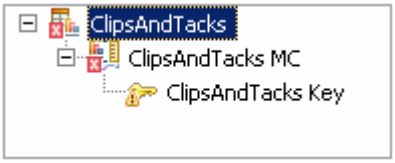
2) Click **Finish**. Click **Yes** over the question dialog if you want to launch the 'Getting Started' information center. The model editor is opened as follows:

Note: When the model editor first opens you will have errors in the model. This is normal and is signaling that required elements are missing or incomplete. Next you will complete these required elements.



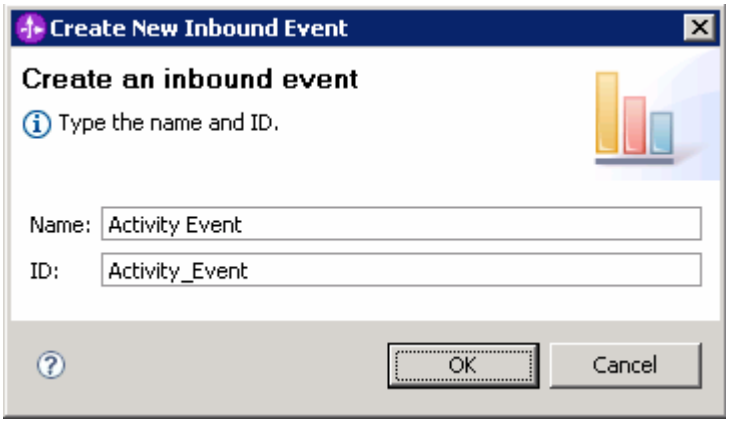
____ 11. Notice in the Monitor Details Model navigator that a new model is created (ClipsAndTacks) along with a monitoring context (ClipsAndTacks MC). Now you need to specify the inbound events that are processed by this model. You need to indicate which event creates a new monitoring context instance and which event terminates the monitoring context instance. You also need to identify the correlation information for the monitoring context so that Monitor knows which monitoring context instance should receive the events. The ClipsAndTacks model uses one event definition, ActivityEvent, to represent the layout of all of the events to be processed. NewOrderEvent derives from ActivityEvent but does not add additional event elements. In ActivityEvent, ActivityName is a field used to identify specific activities in the ClipsAndTacks process. ActivityName is used to determine when to terminate the monitoring context. You will create correlation that is based on orderNumber, which is another field defined in ActivityEvent.

____ 12. In the Monitor Details Model, expand ClipsAndTacks in the navigation view. Then expand ClipsAndTacks MC. You will see that a key has been created for you automatically when you created the model.



____ 13. In the model navigation view, right click on ClipsAndTacks MC, then select **New → Inbound Event** from the pop-up menu

__ a. For the name, type **Activity Event**. Notice that the ID will automatically be set for you to the same name with underscores (**Activity_Event**)



__ b. Click **OK**. The inbound 'ActivityEvent' opens in an editor

__ c. In the inbound 'ActivityEvent' editor, click the '**Add**' button for '**Event Parts**' to create a new event part type in the 'Event Type Details' section. The 'Create New Event part Type' panel opens

__ d. In the 'Create New Event part Type' panel, enter the following:

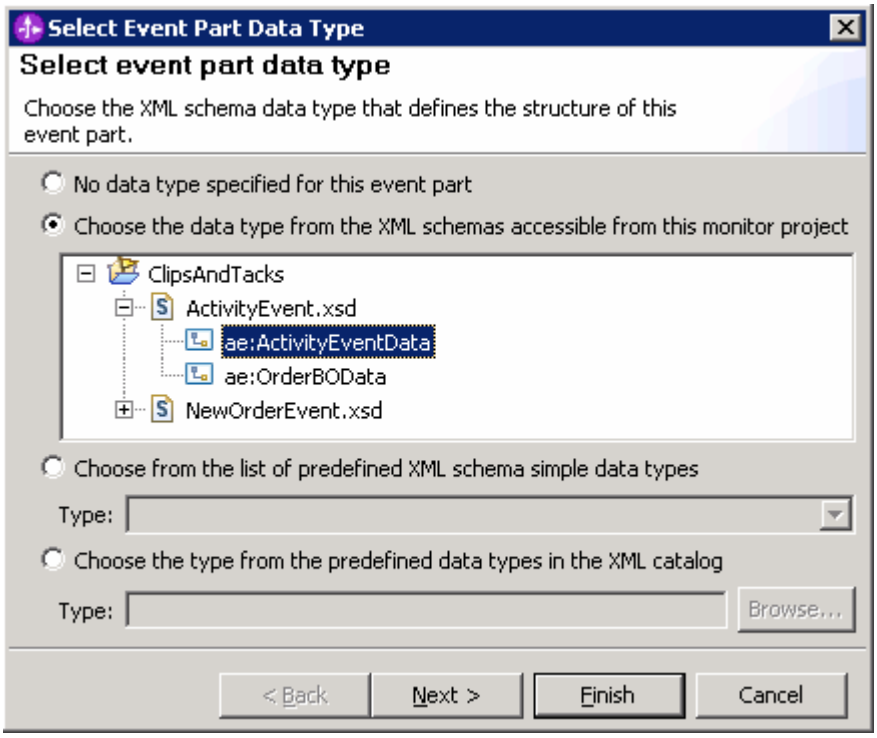
1) Name : **My Event Part1**

2) ID : **My_Event_Part1**

3) Click the '**Select Type**' button for the 'Type' field. The 'Select Event Part Data Type' panel opens

a) Select the check box for 'Choose the data type from XML accessible from this monitor project'

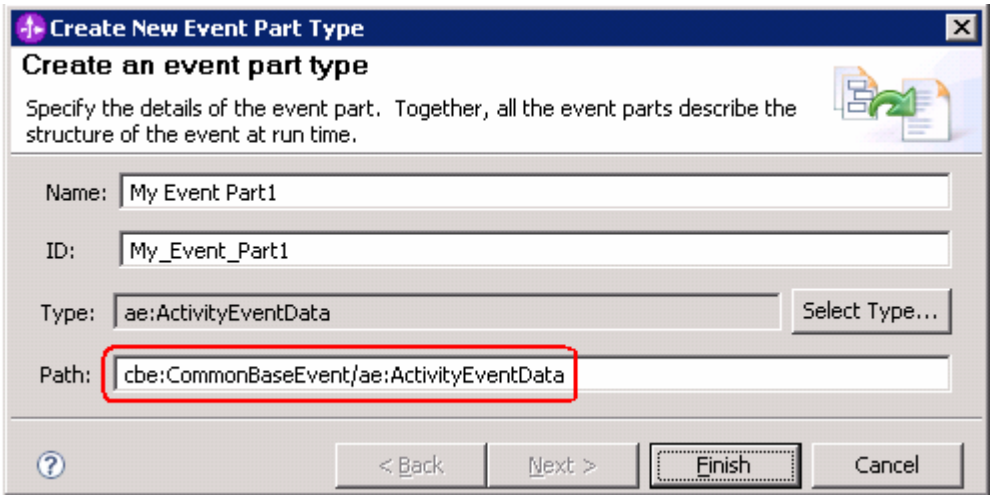
b) Expand **ClipsAndTacks → ActivityEventData** and select '**ae:ActivityEventData**'



c) Click **Finish**

4) Path : **cbe:CommonBaseEvent/ae:ActivityEventData**

Note: The path is an XML Path Language (XPath) expression that identifies the location in the event definition of the structure defined by the event part type. The expression is based on the structure of the actual event instance to be received at run time. It always starts with `cbe:CommonBaseEvent` and navigates into content that is placed in the `xs:any` slot of the Common Base Event. In general, if an XML Schema Definition (XSD) used to define an event structure contains an <xsd:any> or <xsd:anyType> slot, you should use an event part to specify the actual structure of the content that will fill that slot at run time. In this event part, you use **cbe:CommonBaseEvent/ae:ActivityEventData** as the path. Refer to '**allXSDevents.xml**' located in the **Labfiles61.zip** file, to see how the XSD based events are being sent during runtime.



5) Click **Finish**

____ 14. In the inbound 'ActivityEvent' editor, click the '**Add**' button '**Event Parts**' again to create another event part type in the 'Event Type Details' section. The 'Create New Event part Type' panel opens

__ a. In the 'Create New Event part Type' panel, enter the following:

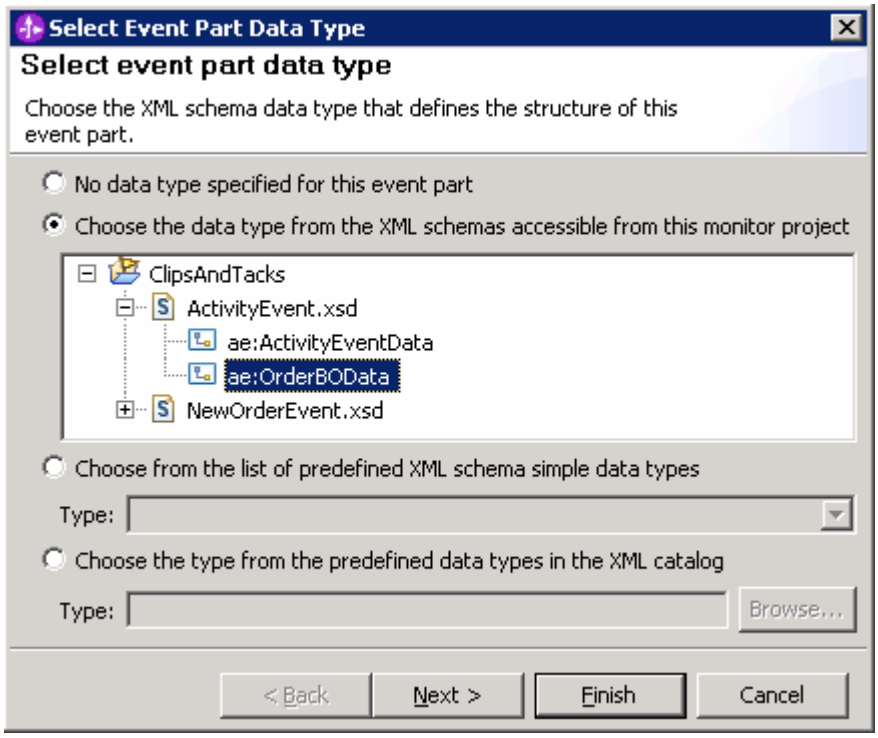
1) Name : **My Event Part2**

2) ID : **My_Event_Part2**

3) Click the '**Select Type**' button for the 'Type' field. The 'Select Event Part Data Type' panel opens

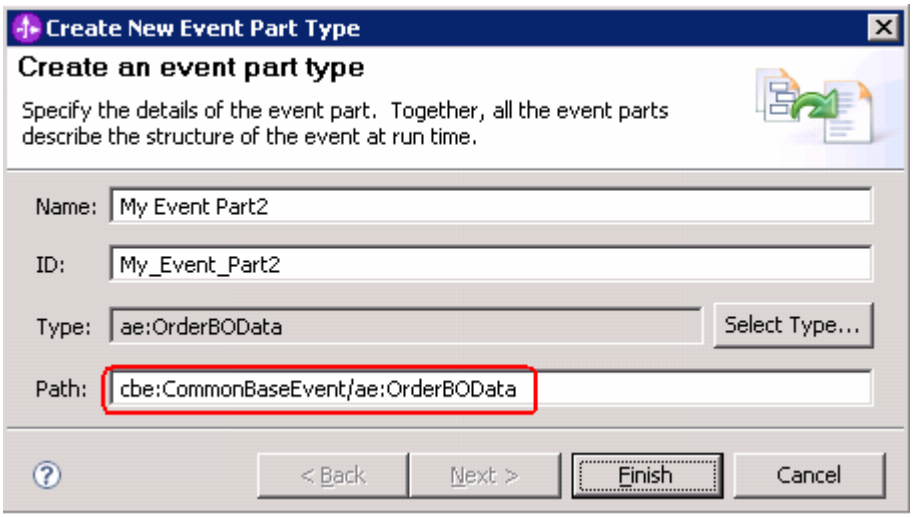
a) Select the check box for 'Choose the data type from XML accessible from this monitor project'

b) Expand **ClipsAndTacks → ActivityEventData.xsd** and select '**ae:OrderBOData**'



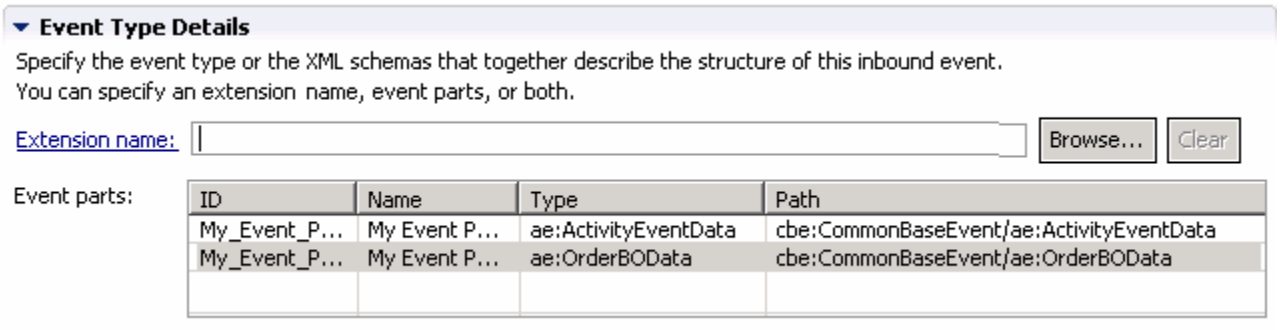
c) Click **Finish**

4) Path : **cbe:CommonBaseEvent/ae:OrderBOData**



5) Click **Finish**

__ b. The final ‘Event Type Details’ section with the two ‘Event parts’ should look like the picture below:



__ c. Save the configuration. **File → Save** or **Ctrl + S**

__ d. To enter the **Filter Condition**, use Content Assist (ctrl-space), or type the expression directly into the expression window. When completed the expression should be as follows. Note the single quotation marks used for string handling.

Activity_Event/My_Event_Part1/ae:businessUnit = 'Clips And Tacks' and
Activity_Event/My_Event_Part1/ae:processName = 'Order Handling'

__ e. To enter the **Correlation Expression**, use Content Assist (ctrl-space), or type the expression directly into the expression window. When complete the expression should be

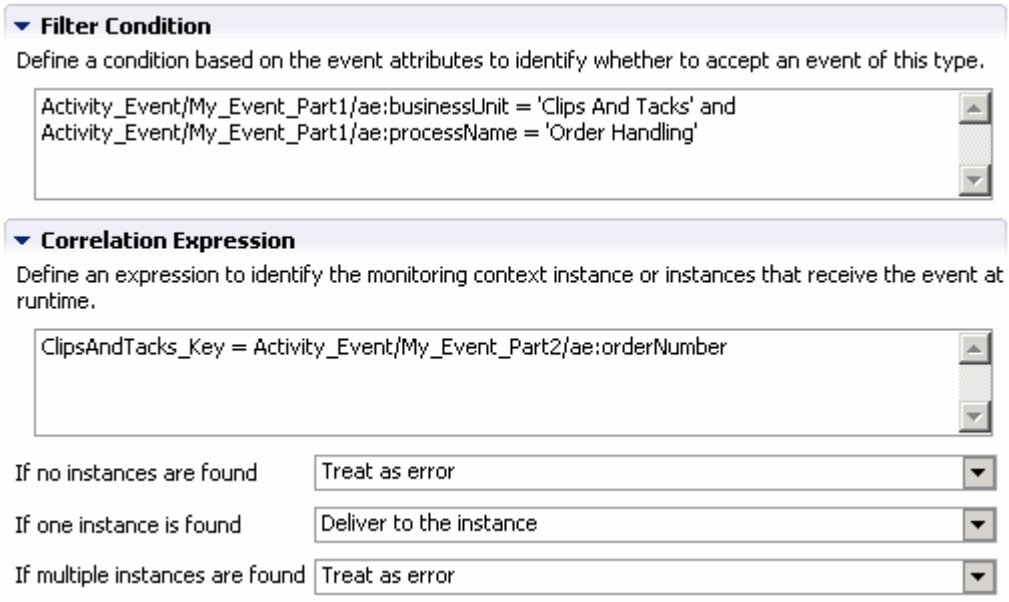
ClipsAndTacks_Key =Activity_Event/My_Event_Part2/ae:orderNumber

__ f. For 'If no instances are found', select 'Treat as error'

__ g. For 'If one instance is found', select 'Deliver to the instance'

__ h. For 'If multiple instances are found', select 'Treat as error'

__ i. Here is a sample of a portion of the definition:



__ j. Save the configuration. **File → Save** or **Ctrl + S**

____ 15. In the model navigation view, right click on ClipsAndTacks MC, then select **New → Inbound Event** from the pop-up menu

__ a. For the name, type **New Order Event**. Notice that the ID will automatically be set for you to the same name with underscores (**New_Order_Event**)

__ b. Click **OK**. The inbound 'NewOrderEvent' opens in an editor

__ c. In the inbound 'NewOrderEvent' editor, click the '**Add**' button for '**Event Parts**' to create a new event part type in the 'Event Type Details' section. The 'Create New Event part Type' panel opens

__ d. In the 'Create New Event part Type' panel, enter the following:

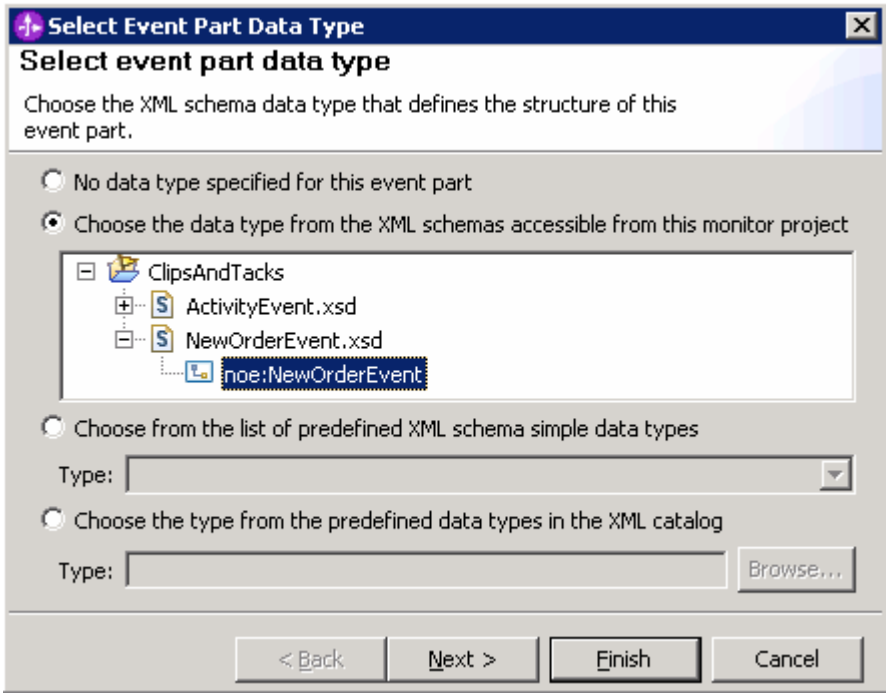
1) Name : **My Event Part**

2) ID : **My_Event_Part**

3) Click the '**Select Type**' button for the 'Type' field. The 'Select Event Part Data Type' panel opens

a) Select the check box for 'Choose the data type from XML accessible from this monitor project'

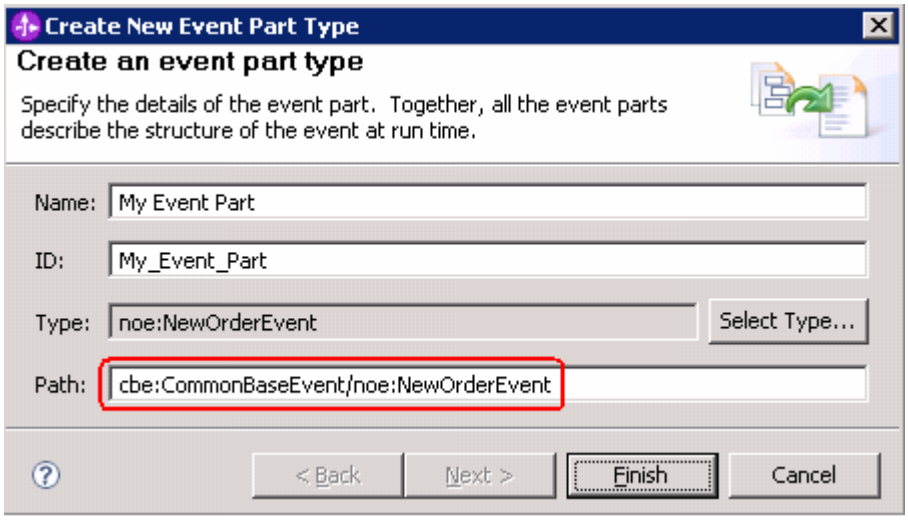
b) Expand **ClipsAndTacks → NewOrderEvent.xsd** and select '**noe:NewOrderEvent**'



c) Click **Finish**

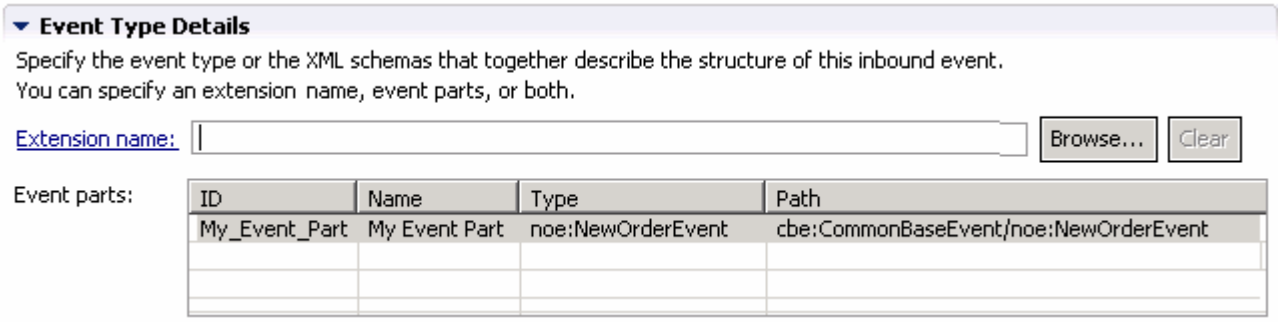
4) Path : **cbe:CommonBaseEvent/noe:NewOrderEvent**

Note: The path is an XML Path Language (XPath) expression that identifies the location in the event definition of the structure defined by the event part type. The expression is based on the structure of the actual event instance to be received at run time. It always starts with `cbe:CommonBaseEvent` and navigates into content that is placed in the `xs:any` slot of the Common Base Event. In general, if an XML Schema Definition (XSD) used to define an event structure contains an <xsd:any> or <xsd:anyType> slot, you should use an event part to specify the actual structure of the content that will fill that slot at run time. In this event part, you use **cbe:CommonBaseEvent/ noe:NewOrderEvent** as the path. Refer to '**allXSDevents.xml**' part of the **Labfiles61.zip** to see how the XSD based events are being sent during runtime.



5) Click **Finish**

__ e. The final 'Event Type Details' section with the 'Event part' should look like the picture below:



__ f. Save the configuration. **File → Save** or **Ctrl + S**

__ g. To enter the Filter Condition, use Content Assist (ctrl-space), or type the expression directly into the expression window. When complete, the expression should be as follows. Note the single quotation marks used for string handling.

New_Order_Event/My_Event_Part/noe:NEWActivityEventData/ae:businessUnit = 'Clips And Tacks' and New_Order_Event/My_Event_Part/noe:NEWActivityEventData/ae:processName ='Order Handling'

__ h. To enter the Correlation Expression, use Content Assist (ctrl-space), or type the expression directly into the expression window. When complete the expression should be

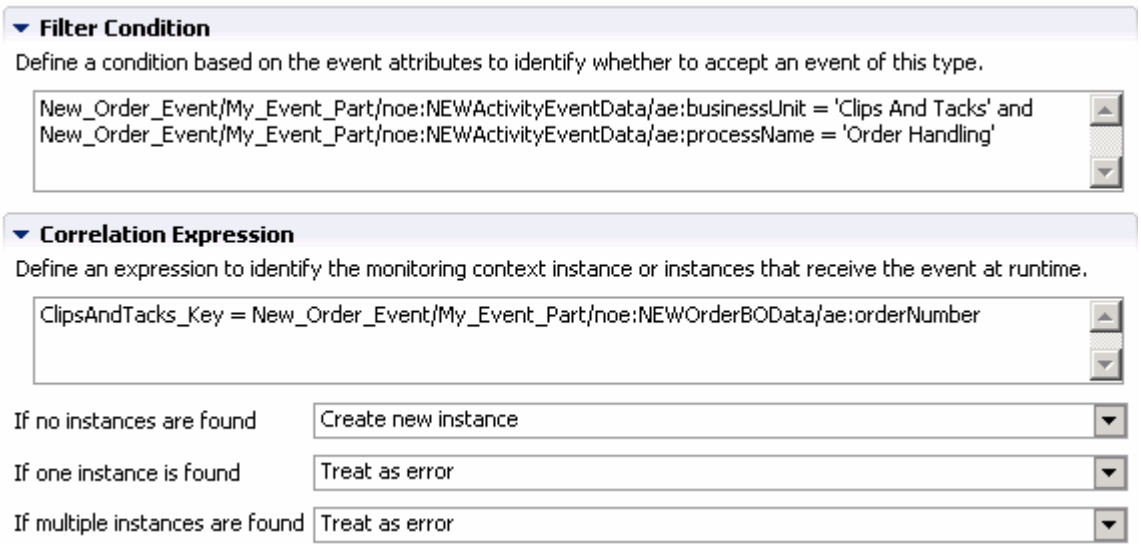
ClipsAndTacks_Key =
New_Order_Event/My_Event_Part/noe:NEWOrderBOData/ae:orderNumber

__ i. For 'If no instances are found', select 'Create new instance'

__ j. For 'If one instance is found', select 'Treat as error'

__ k. For 'If multiple instances are found', select 'Treat as error'

__ l. Here is a sample of a portion of the definition:



__ m. Save the configuration. **File → Save** or **Ctrl + S**

____ 16. In the model navigation view, double click **ClipsAndTacks Key**, to open the key in the editor

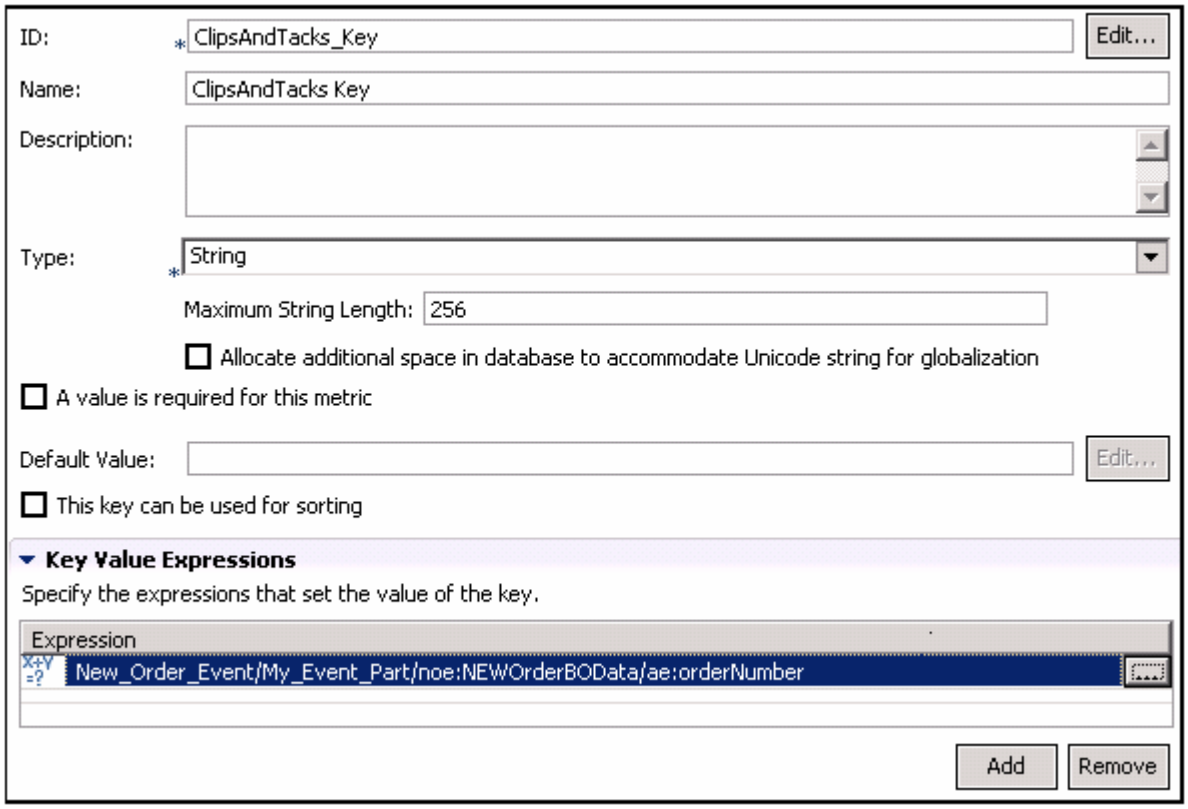
__ a. For Key Value Expressions, click '**Add…**' Note that double clicking in the table will produce the same results.

__ b. A row is added to the table and in the expression cell it shows << No expression specified >>. Click on this cell of the table, then a button is displayed. Click on this button and the expression editor is displayed. You can use Content Assist, or type this expression directly into the window:

New_Order_Event/My_Event_Part/noe:NEWOrderBOData/ae:orderNumber

__ c. Click **OK**.

__ d. Here is a sample of the key:



__ e. Save the configuration. **File → Save** or **Ctrl + S**

Note: At this point there should not be any errors remaining in the model. The errors shown initially were the result of the monitoring context key not being complete.

____ 17. Create triggers that indicate the end of the monitoring context. The first trigger is based on an order being shipped, and the second trigger is based on an order being cancelled.

__ a. In the model navigation view, right click on **ClipsAndTacks MC**, then select **New → Trigger**. Enter the following values:

1) Name : **Ship Order to Customer Trigger**

2) ID : **Ship_Order_to_Customer_Trigger**

3) Select the check box next to '**Terminate monitoring context**'

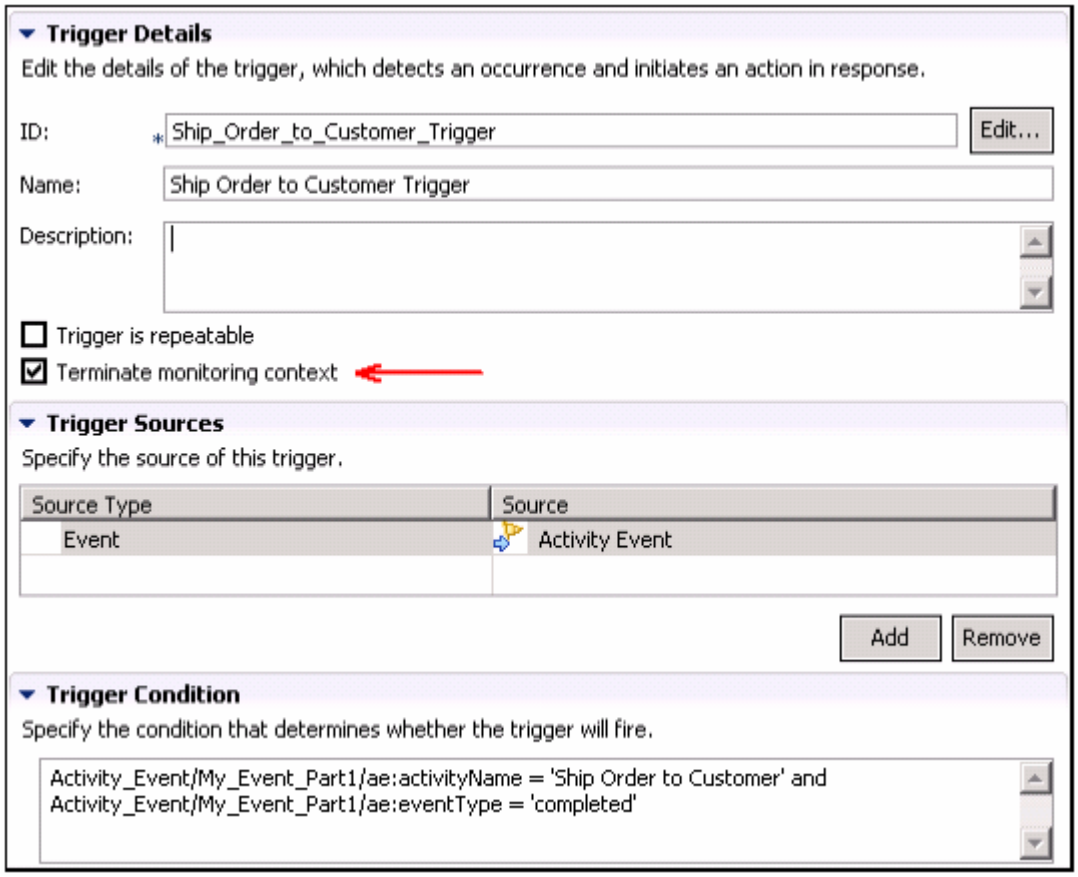
4) For Trigger Sources, click Add, then select **Other source type → Activity Event** and then click OK

5) For Trigger Condition, enter

Activity_Event/My_Event_Part1/ae:activityName = 'Ship Order to Customer' and
Activity_Event/My_Event_Part1/ae:eventType = 'completed'

__ b. Save the configuration. **File → Save** or **Ctrl + S**

__ c. Here is a sample of this trigger:



__ d. In the model navigation view, right click on ClipsAndTacks MC, then select **New > Trigger**. Enter these values:

1) Name : **Cancel Trigger**

2) ID : **Cancel_Trigger**

3) Select the check box next to '**Terminate monitoring context**'

4) For Trigger Sources, click Add, then select Other source type > Activity Event, then click OK.

5) For Trigger Condition, enter

Activity_Event/My_Event_Part1/ae:activityName = 'Cancel Order and Send Notification' and
Activity_Event/My_Event_Part1/ae:eventType = 'completed'

__ e. Save the configuration. **File → Save** or **Ctrl + S**

____ 18. Create a KPI for Average fulfillment 3 days or less. First you will create a trigger which fires when a new order is started. Metrics are created to hold the order start time and order end time. The values of the metrics are set when the new order trigger fires. You create a duration metric which is calculated based on subtracting the order end time metric from the order start time metric. You will create a KPI context which is a container to hold the KPIs. Then you create a KPI which is based on the duration metric and you apply an average function to it.

__ a. In the model navigation view, right click on ClipsAndTacks MC, then select **New → Trigger**. Enter these values:

1) Name : **New Order Trigger**

2) ID : **New_Order_Trigger**

3) For Trigger Sources, click Add, then select Other source type > New Order Event, then click OK.

4) For Trigger condition, enter

New_Order_Event/My_Event_Part/noe:NEWActivityEventData/ae:activityName = 'Check Order Handling Policy for Automatic Approval' and
New_Order_Event/My_Event_Part/noe:NEWActivityEventData/ae:eventType = 'started'

5) Save the configuration. **File → Save** or **Ctrl + S**

__ b. In the model navigation view, right click on ClipsAndTacks MC, then select **New → Metric**. Enter these values:

1) Name : **Order Start Time**

2) ID : **Order_Start_Time**

3) Type : **DateTime**

4) For Metric Value Expressions, click Add

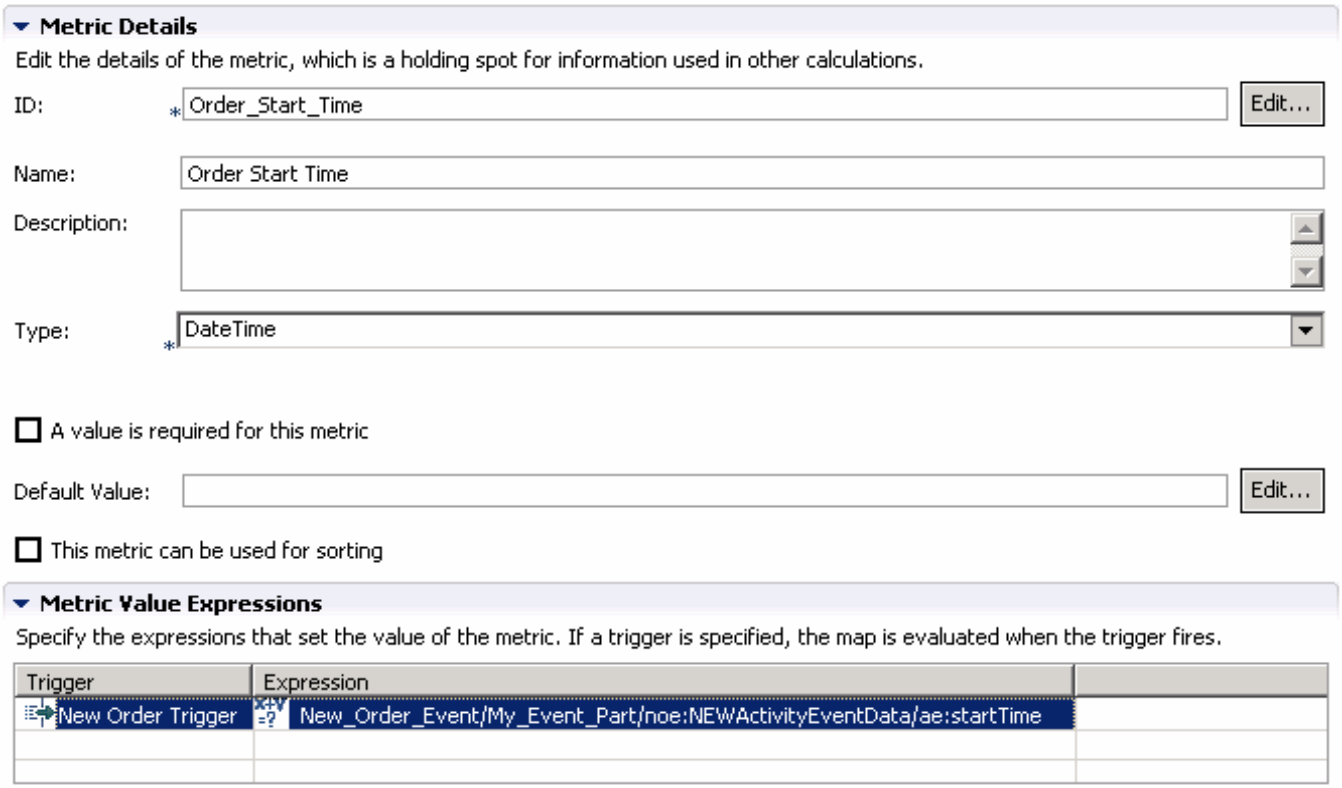
5) In the first row of the Metric Value Expressions table, select the cell under column Trigger, then a button is shown, then click the button and select Trigger type > ClipsAndTacks MC > New Order Trigger, click OK

6) In the first row of the Metric Value Expressions table, select the cell under column Expression, then a button is shown, then click the button and enter this text:

New_Order_Event/My_Event_Part/noe:NEWActivityEventData/ae:startTime

7) Click OK.

8) Save the configuration. **File → Save** or **Ctrl + S**



__ c. In the model navigation view, right click on ClipsAndTacks MC, then select **New → Metric**. Enter these values:

1) Name : **Order End Time**

2) ID : **Order_End_Time**

3) Type : **DateTime**

4) For Metric Value Expressions, click Add

5) In the first row of the Metric Value Expressions table, select the cell under column Trigger, then a button is shown, then click the button and select Trigger type > ClipsAndTacks MC > Ship Order to Customer Trigger, click OK

6) In the first row of the Metric Value Expressions table, select the cell under column Expression, then a button is shown, then click the button and enter this text:

Activity_Event/My_Event_Part1/ae:endTime

7) Click OK.

8) For Metric Value Expressions, click Add

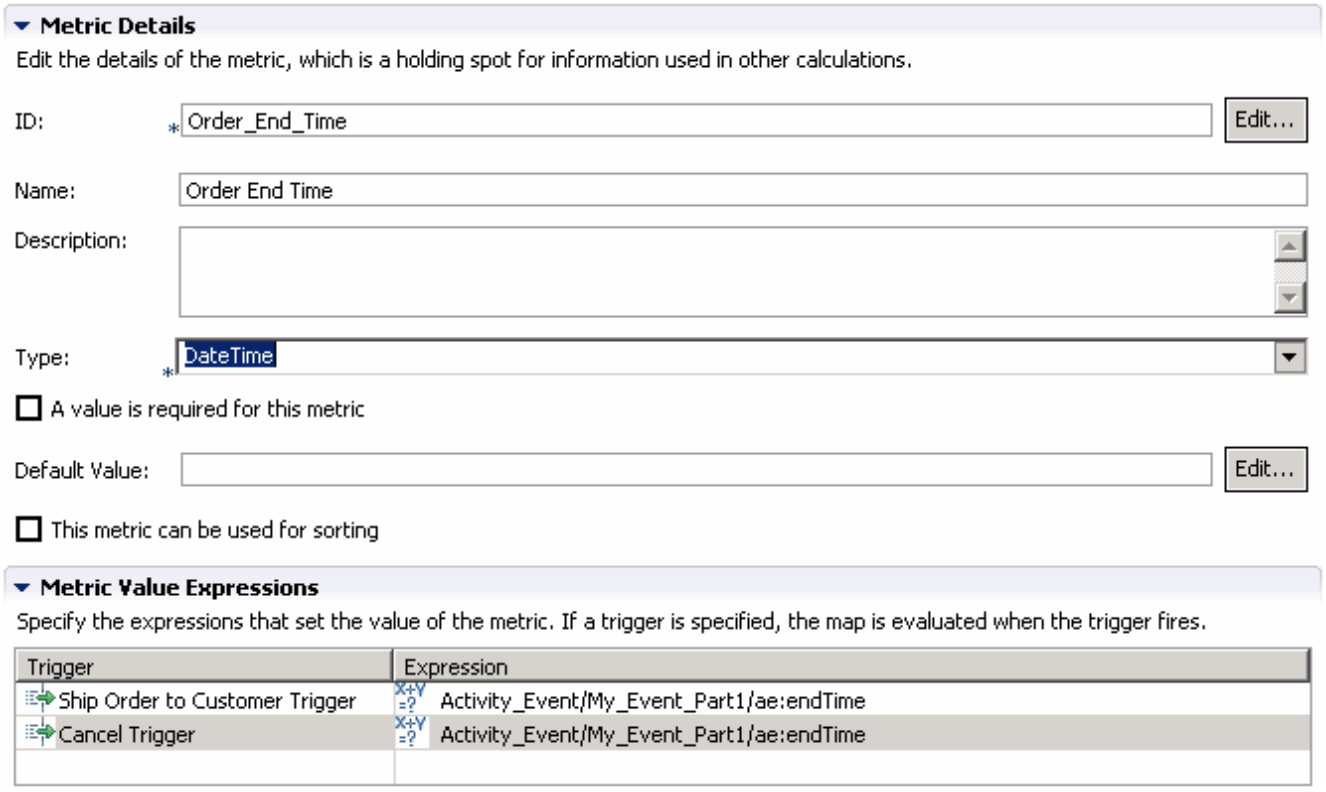
9) In the second row of the Metric Value Expressions table, select the cell under column Trigger, then a button is shown, then click the button and select Trigger type > ClipsAndTacks MC > Cancel Trigger, click OK

10) In the second row of the Metric Value Expressions table, select the cell under column Expression, then a button is shown, then click the button and enter this text:

Activity_Event/My_Event_Part1/ae:endTime

11) Click OK.

__ d. Save the configuration. **File → Save** or **Ctrl + S**



__ e. In the model navigation view, right click on ClipsAndTacks MC, then select **New > Metric**. Enter these values:

1) Name : **Order Fulfillment Duration**

2) ID : **Order_Fulfillment_Duration**

3) Type : **Duration**

4) For Default Value, click Edit…

5) On the Select Duration dialog, click OK, and this will set the default duration to zero

6) For Metric Value Expressions, click Add

7) In the first row of the Metric Value Expressions table, select the cell under column Trigger, then a button is shown, then click the button and select Trigger type > ClipsAndTacks MC > Ship Order to Customer Trigger, click OK

8) In the first row of the Metric Value Expressions table, select the cell under column Expression, then a button is shown, then click the button and enter this text:

xs:dayTimeDuration(Order_End_Time - Order_Start_Time)

9) Click OK

10) For Metric Value Expressions, click Add

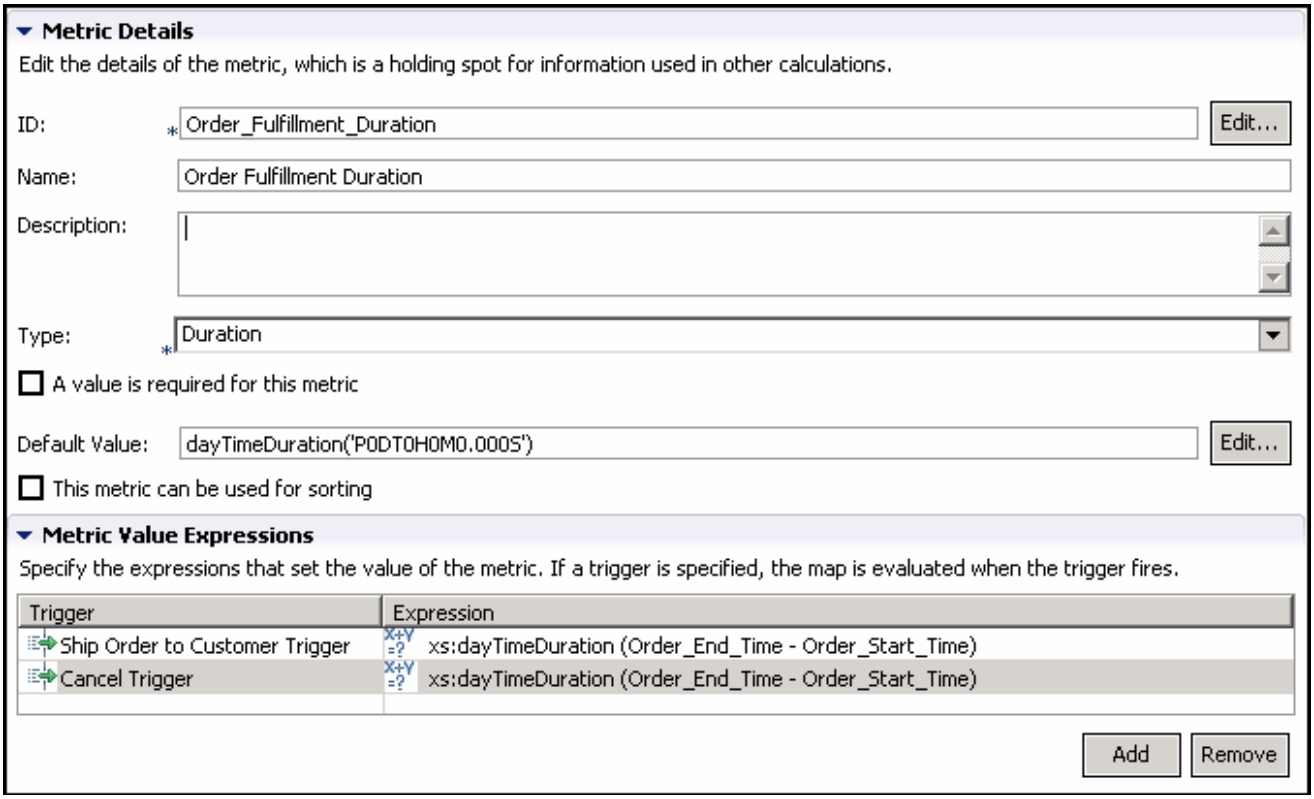
11) In the second row of the Metric Value Expressions table, select the cell under column Trigger, then a button is shown, then click the button and select Trigger type > ClipsAndTacks MC > Cancel Trigger, click OK

12) In the second row of the Metric Value Expressions table, select the cell under column Expression, then a button is shown, then click the button and enter this text:

xs:dayTimeDuration (Order_End_Time - Order_Start_Time)

13) Click OK.

__ f. Press **Ctrl-S** to save your work.



__ g. Click the **KPI Model tab**.

__ h. Create a KPI context to store the KPIs.

1) In the navigation view of the KPI Model, right click on ClipsAndTacks, then select **New → KPI Context**

a) Name :- **My KPI Context**

b) ID :- **My_KPI_Context**

__ i. Create the average order fulfillment KPI.

Note that you will create the KPI based on events for August 2006. This will work for you no matter what month or year that you run this lab because the timestamps in the test data are based on August 2006, so your results will be the same as in these lab instructions.

1) In the navigation view of the KPI Model, right click on **My KPI Context**, then select **New → KPI**

   a) Name :- **Average Order Fulfillment KPI August 2006**

   b) ID :- **Average_Order_Fulfillment_KPI_August_2006**

2) Enter these values:

   a) Type – Duration

   b) In KPI Target and Ranges, for Target click **Details**…, then change the value to 3 Days, then click OK

   c) In KPI Target and Ranges, for Ranges select Actual value

   d) In the Range table,

      (1) Click Add, or double click anywhere in the table

      (2) Change the Name to Day 1, and the ID to Day_1

      (3) A row is created in the Range table with a range name Day 1.

      (4) For this row, select the cell in the Start value column and a button is displayed. Click this button, then change all values to 0, then click OK.

      (5) For this row, select the cell in the End value column and a button is displayed. Click this button, then change Days to 1 and the other values to 0, then click OK.

   e) Now you have created one range called Day 1. Repeat the above step to create four more ranges:

      (1) Day 2 with start values 1 Days and end value 2 Days

      (2) Day 3 with start values 2 Days and end value 3 Days

      (3) Day 4 with start values 3 Days and end value 4 Days

      (4) Day 5 with start values 4 Days and end value 5 Days

   f) KPI Value – select '**Base this KPI on a metric and an aggregation function**'

   g) Under KPI Details, for Monitoring context, click Browse…, then select '**ClipsAndTacks MC**', then click OK

   h) Metric – click Browse…, then select **Order Fulfillment Duration**, then click OK

   i) Aggregation function – select **Average**.

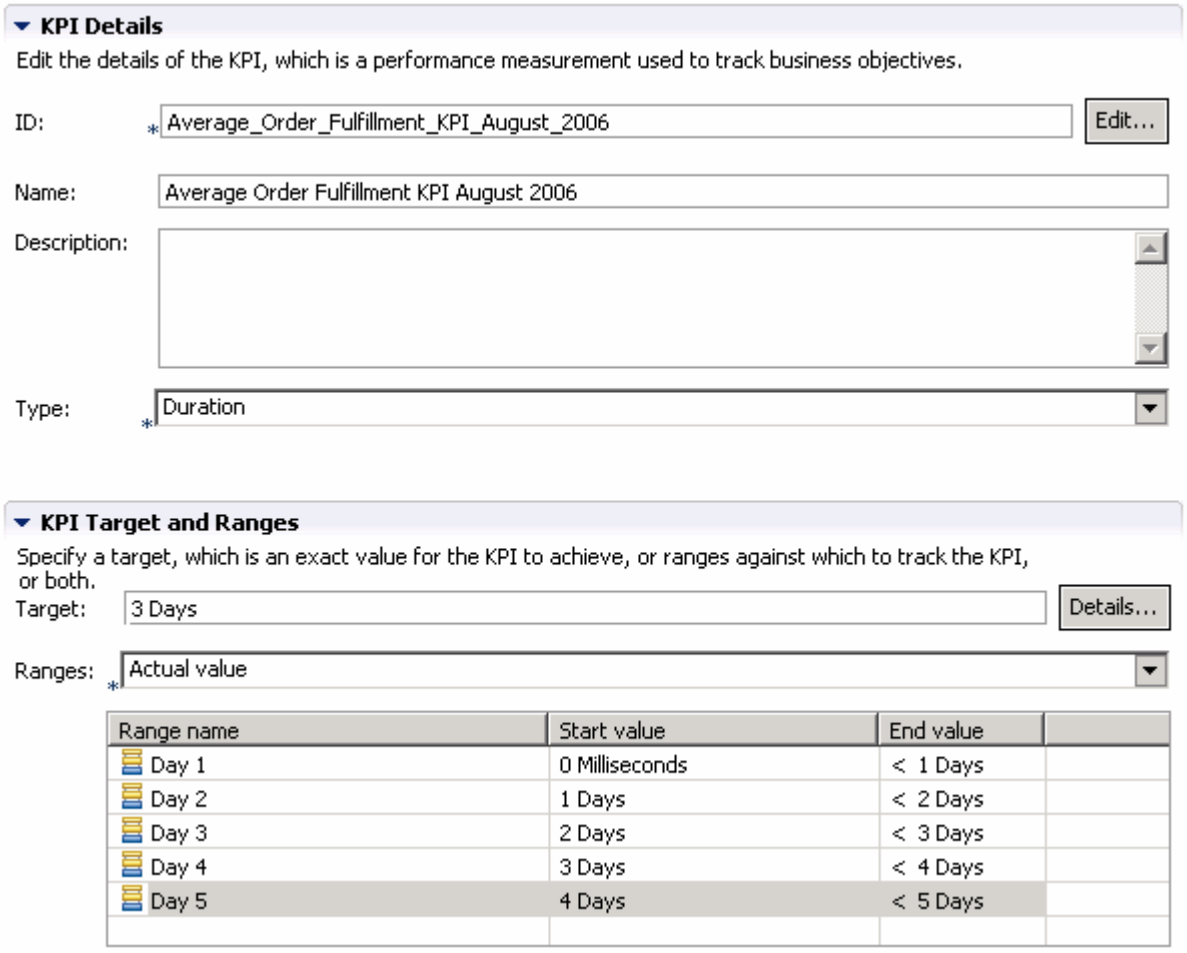
   j) Under Time Filter, for Metric, click Browse…, then select **Order Start Time**, then click OK

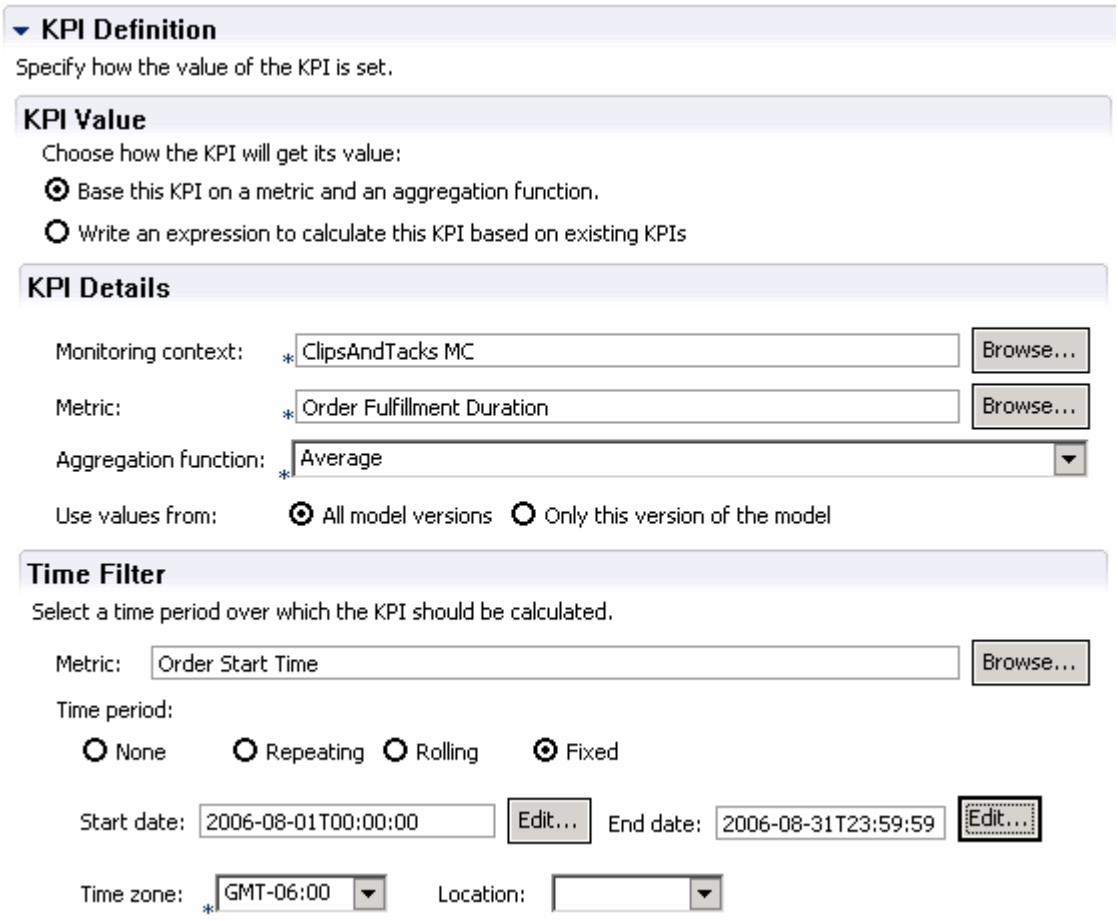
k) Specify time period – **Fixed**

l) Start date – click Edit... then select date **2006-08-01**, time **00:00:00**, click OK

m) End date – click Edit…, then select **2006-08-31**, time **23:59:59**, click OK

3) Press **Ctrl-S** to save your work.





__ j. Note that this KPI as it stands averages order fulfillment time for shipped orders and cancelled orders, but you probably are not interested in the cancelled orders to be included in this KPI. So update the average order fulfillment KPI to show the values for shipped orders only. First you will create a new metric to hold the order status. Then you will add this metric as a data filter on the KPI.

__ k. Click the **Monitor Details Model tab**.

__ l. In the Monitor Details Model navigation view, right click on ClipsAndTacks MC, then select **New → Metric**. Enter these values:

1) Name :- **Order Status**

2) ID :- **Order_Status**

3) Type :- **String**

4) Select the check box for '**A value is required for this metric**'

5) Default Value (enter this text with single quotation marks) – '**New**'

6) For Metric Value Expressions, click **Add**

7) In the first row of the Metric Value Expressions table, select the cell under column Trigger, then a button is shown, then click the button and select Trigger type > ClipsAndTacks MC > Cancel Trigger, click OK

8) In the first row of the Metric Value Expressions table, select the cell under column Expression, then a button is shown, then click the button and enter this text with the quotation marks:

‘Cancelled’

9) Click OK.

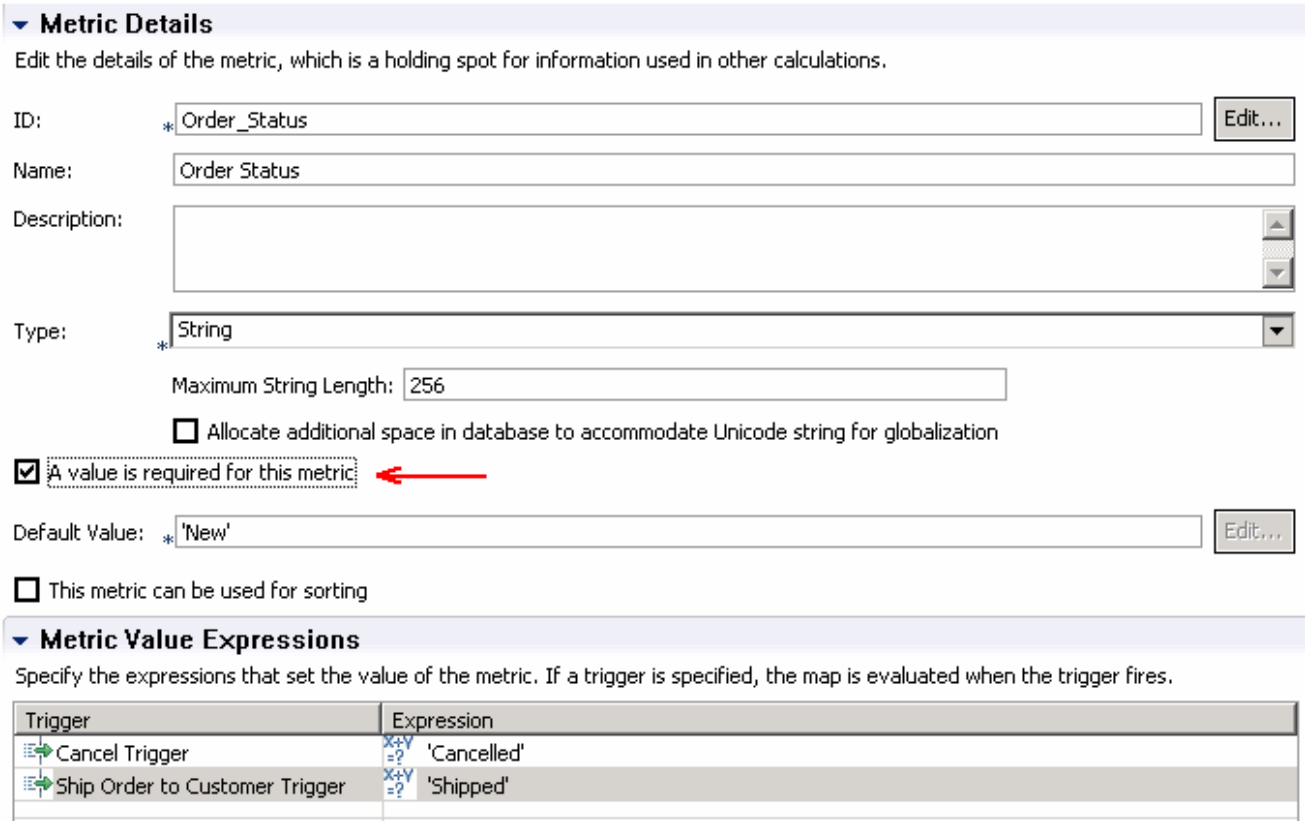
10) For Metric Value Expressions, click Add

11) In the second row of the Metric Value Expressions table, select the cell under column Trigger, then a button is shown, then click the button and select Trigger type > ClipsAndTacks MC > Ship Order to Customer Trigger, click OK

12) In the second row of the Metric Value Expressions table, select the cell under column Expression, then a button is shown, then click the button and enter this text with the quotation marks:

‘Shipped’

13) Press **Ctrl-S** to save your work



__ m. Update the order fulfillment KPI to use the order status as a filter.

1) Click the **KPI Model tab**.

2) In the KPI Model navigation view, navigate to ClipsAndTacks > My KPI Context > Average Order Fulfillment KPI August 2006

3) In the Data Filter section for this KPI, click Add

4) In the dialog, navigate to **ClipsAndTacks MC → Order Status**, then click OK

5) In the row containing metric Order Status, make sure the operation defaults to 'equals'. Also, click the cell in the column Values, then enter this text (with quotation marks):

'Shipped'

6) Press **Ctrl-S** to save your work.

____ 19. Create a KPI for number of approved orders greater than 90%. First you will create a KPI to count the number of total orders. Then you will create a KPI to count the number of shipped orders. Finally, you will create a KPI that calculates the approval percentage as this formula: (ship count / order count) * 100.

__ a. Click the **KPI Model tab** then create the order count KPI.

1) In the navigation view of the KPI Model, right click on My KPI Context, then select **New > KPI**

a) Name :- **Order Count KPI**

b) ID :- **Order_Count_KPI**

2) Enter these values:

a) Type :- **Decimal**

b) Do not specify targets nor ranges

c) KPI Value – select '**Base this KPI on a metric and an aggregation function**'

d) Under KPI Details, for Monitoring context, click Browse…, then select '**ClipsAndTacks MC**', then click OK

e) Metric – click Browse…, then select **ClipsAndTacks Key**, then click OK

f) Aggregation function – select **Count**

3) Press **Ctrl-S** to save your work.

__ b. Create the ship count KPI.

1) In the navigation view of the KPI Model, right click on My KPI Context, then select **New > KPI**

a) Name :- **Ship Count KPI**

b) ID :- **Ship_Count_KPI**

2) Enter these values:

a) Type :- **Decimal**

b) Do not specify targets nor ranges

c) KPI Value – select '**Base this KPI on a metric and an aggregation function**'

d) Under KPI Details, for Monitoring context, click Browse…, then select '**ClipsAndTacks MC**', then click OK

e) Metric – click Browse…, then select **ClipsAndTacks Key**, then click OK

f) Aggregation function – select **Count**

g) In the Data Filter section for this KPI, click **Add**

h) In the dialog, navigate to **ClipsAndTacks MC → Order Status**, then click OK

i) In the row containing metric Order Status, make sure the operation defaults to 'equals'. Also, click the cell in the column Values, then enter this text (with quotation marks):

'Shipped'

3) Press **Ctrl-S** to save your work.

__ c. Create the percent of orders approved KPI.

1) In the navigation view of the KPI Model, right click on My KPI Context, then select **New > KPI**

a) Name :- **Percent of Orders Approved KPI**

b) ID :- **Percent_of_Orders_Approved_KPI**

2) Enter these values:

a) Type :- **Decimal**

b) In KPI Target and Ranges, for Target click Details…, then change the value to **90**, then click OK

c) In KPI Target and Ranges, for Ranges select **Actual value**

d) In the Range table,

(1) Click **Add**

(2) Change the Name to **Low Range**, and the ID to **Low_Range**

(3) A row is created in the Range table with a range name Low Range.

(4) For this row, select the cell in the Start value column and a button is displayed. Click this button, then change the value to 0 and then click OK.

(5) For this row, select the cell in the End value column and a button is displayed. Click this button, then change the value to 90 and then click OK.

e) Repeat the above step to create one more range:

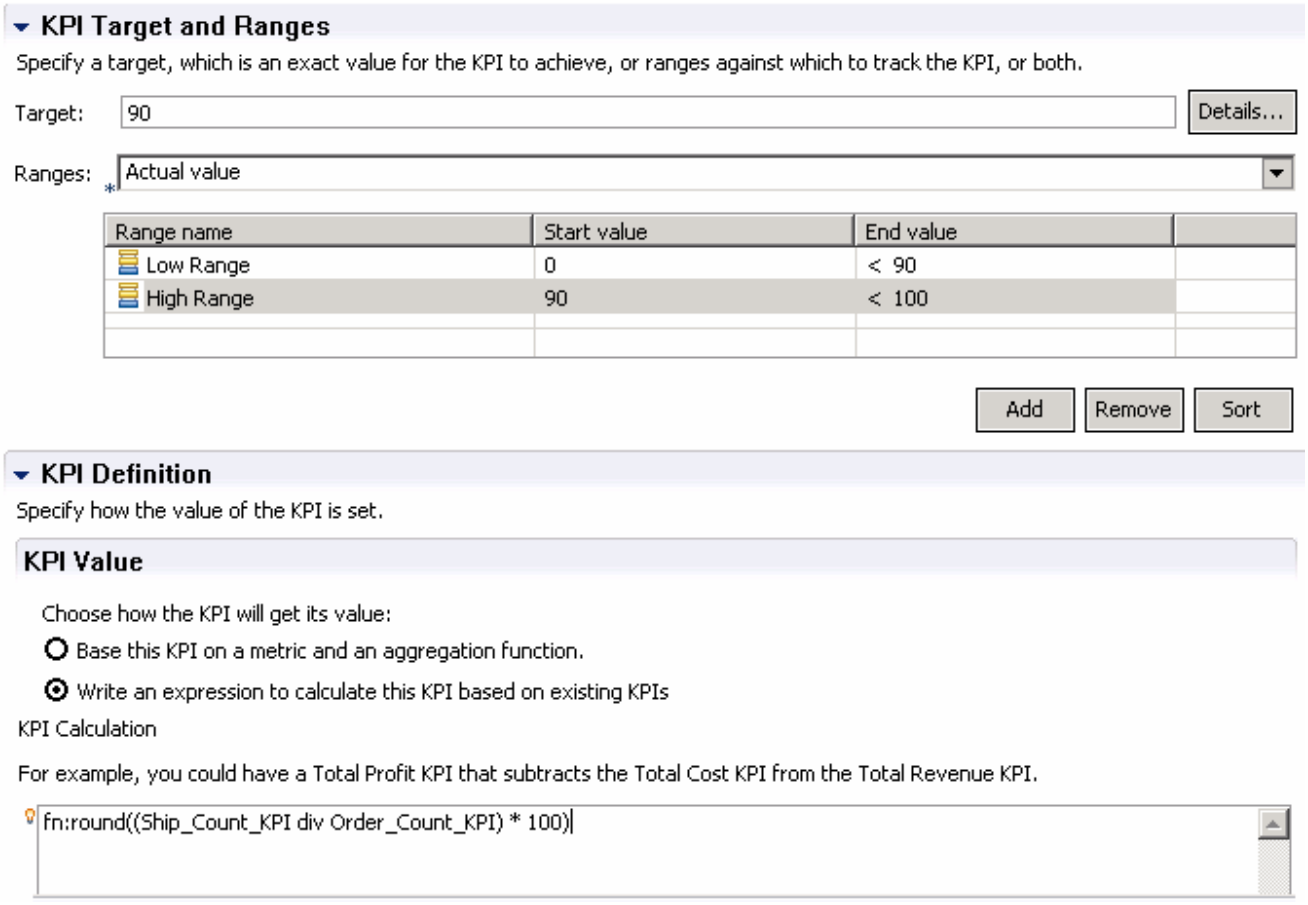
(1) **High Range** with start value 90 and end value 100

f) KPI Value – select '**Write an expression to calculate this KPI based on existing KPIs**'

g) For KPI Calculation, enter this text:

fn:round((Ship_Count_KPI div Order_Count_KPI) * 100)

3) Press **Ctrl-S** to save your work.



____ 20. In the dashboards, you might like to see the total order price, the average order price and the total number of orders. And you might be interested in seeing this information by country and city, and by order status. So now you will create dimensions and measures that allow you to see this information. For aggregated numeric information like total order price, you will create a measure in the dimensional model. For textual attributes like order status, you will create a dimension in the dimensional model. In order to see the country and city information, you will create a multi-level location dimension containing country as the first level and city as the second level. Each measure and dimension needs a metric as a source, so you will also need to create metrics for country, city and total price.

__ a. Click on the **Dimensional Model tab**.

__ b. Add order status as a dimension to the Dimensional Model.

1) In the navigation view, click on ClipsAndTacks MC Cube

2) In the dimensions table, click **New Dimension…**

a) Name :- **Order Status Dimension**

b) ID :- **Order_Status_Dimension**

c) Click **OK**

3) Click **New Level…**

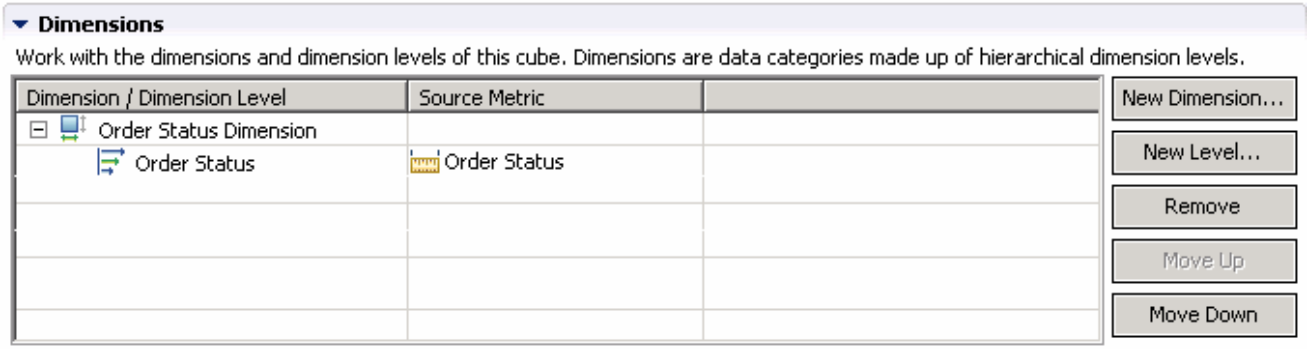
a) Name :- **Order Status**

b) ID :- **Order_Status**

c) Source metric – Click Browse…, then select **ClipsAndTacks MC → Order Status**. Then click OK.

d) Click OK.

4) Press **Ctrl-S** to save your work.



__ c. Click the **Monitor Details Model tab**.

__ d. In the Monitor Details Model navigation view, right click on ClipsAndTacks MC, then select **New > Metric**. Enter these values:

1) Name :- **country**

2) ID :- **country**

3) Type :- **String**

4) Select the check box for '**A value is required for this metric**'

5) Default value : **''** (**Note:** Empty single quotation marks)

6) For Metric Value Expressions, click Add

7) In the first row of the Metric Value Expressions table, select the cell under column Trigger, then a button is shown, then click the button and select Trigger type > ClipsAndTacks MC > New Order Trigger, click OK

8) In the first row of the Metric Value Expressions table, select the cell under column Expression, then a button is shown, then click the button and enter this text:

New_Order_Event/My_Event_Part/noe:NEWOrderBOData/ae:country

9) Click OK.

10) Press **Ctrl-S** to save your work

__ e. In the Monitor Details Model navigation view, right click on ClipsAndTacks MC, then select **New > Metric**. Enter these values:

1) Name :- **city**

2) ID :- **city**

3) Type :- **String**

4) Select the check box for '**A value is required for this metric**'

5) Default value : **''** (**Note:** Empty single quotation marks)

6) For Metric Value Expressions, click Add

7) In the first row of the Metric Value Expressions table, select the cell under column Trigger, then a button is shown, then click the button and select Trigger type > ClipsAndTacks MC > New Order Trigger, click OK

8) In the first row of the Metric Value Expressions table, select the cell under column Expression, then a button is shown, then click the button and enter this text:

New_Order_Event/My_Event_Part/noe:NEWOrderBOData/ae:city

9) Click OK.

10) Press **Ctrl-S** to save your work

__ f. In the Monitor Details Model navigation view, right click on ClipsAndTacks MC, then select **New > Metric**. Enter these values:

1) Name :- **totalPrice**

2) ID :- **totalPrice**

3) Type :- **Decimal**

4) For Metric Value Expressions, click Add

5) In the first row of the Metric Value Expressions table, select the cell under column Trigger, then a button is shown, then click the button and select Trigger type > ClipsAndTacks MC > New Order Trigger, click OK

6) In the first row of the Metric Value Expressions table, select the cell under column Expression, then a button is shown, then click the button and enter this text:

New_Order_Event/My_Event_Part/noe:NEWOrderBOData/ae:totalPrice

7) Click OK.

8) Press **Ctrl-S** to save your work

__ g. Click on the **Dimensional Model tab**.

__ h. Add location as a dimension to the Dimensional Model.

1) In the navigation view, click on ClipsAndTacks MC Cube

2) In the dimensions table, click **New Dimension…**

a) Name :- **Location**

b) ID :- **Location**

c) Click **OK**

3) Click **New Level…**

a) Name :- **Country**

b) ID :- **Country**

c) Source metric – Click Browse…, then select **ClipsAndTacks MC → country**. Then click OK.

d) Click OK.

4) Click **New Level…**

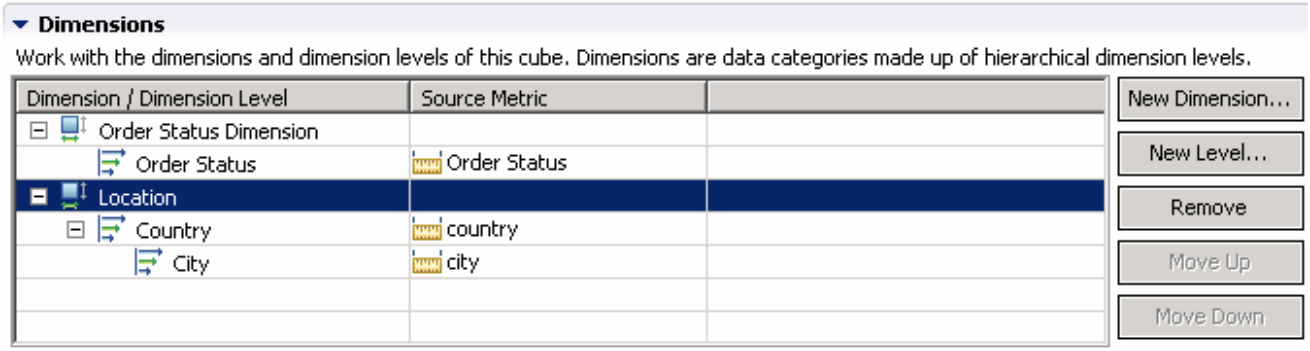
a) Name :- **City**

b) ID :- **City**

c) Source metric – Click Browse…, then select **ClipsAndTacks MC → city**. Then click OK.

d) Click OK.

5) Press **Ctrl-S** to save your work.



__ i. Ensure that the **Dimensional Model tab** is selected.

__ j. In the Dimensional Model, add a measure for average order price.

1) In the navigation view, click on ClipsAndTacks MC Cube

2) In the Measures table, click **New …**

a) Name :- **Average Order Price**

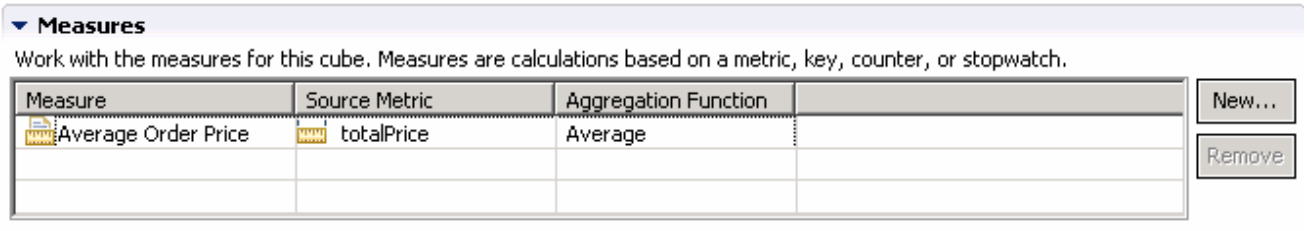
b) ID :- **Average_Order_Price**

c) Source metric - Click Browse…, then select **ClipsAndTacks MC → totalPrice**.

d) Aggregation function: - select **Average**.

e) Then click OK.

3) Press **Ctrl-S** to save your work.



__ k. In the Dimensional Model, add a measure for sum of order price.

1) In the navigation view, click on ClipsAndTacks MC Cube

2) In the Measures table, click **New …**

a) Name :- **Sum Order Price**

b) ID :- **Sum_Order_Price**

c) Source metric - Click Browse…, then select **ClipsAndTacks MC → totalPrice**.

d) Aggregation function – select **Sum**.

e) Then click OK.

3) Press **Ctrl-S** to save your work.

__ l. In the Dimensional Model, add a measure for order count.

1) In the navigation view, click on ClipsAndTacks MC Cube

2) In the Measures table, click **New …**

a) Name :- **Order Count**

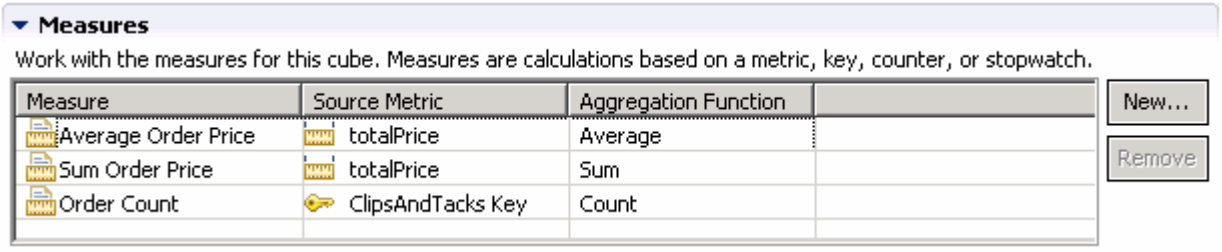
b) ID :- **Order_Count**

c) Source metric - Click Browse…, then select **ClipsAndTacks MC → ClipsAndTacks Key**.

d) Aggregation function – select **Count**.

e) Then click OK.

3) Press **Ctrl-S** to save your work.



____ 21. In the dashboards you might like to see the average order price and track that relative to a target.

__ a. Create a KPI for the average order price.

1) Click the **KPI Model tab**.

2) In the navigation view of the KPI Model, right click on My KPI Context, then select **New → KPI**

a) Name :- **Average Order Price KPI (Dollars)**

b) ID :- **Average_Order_Price_KPI__x0028_Dollars_x0029_**

3) Enter these values:

a) Type :- **Decimal**

b) In KPI Target and Ranges, for Target click Details…, then change the value to **300**, then click OK

c) In KPI Target and Ranges, for Ranges select **Actual value**

d) In the Range table,

(1) Click **Add**

(2) Change the Name to **Low Range**, and the ID to **Low_Range**

(3) A row is created in the Range table with a range name Low Range.

(4) For this row, select the cell in the Start value column and a button is displayed. Click this button, then change the value to 0, and then click OK.

(5) For this row, select the cell in the End value column and a button is displayed. Click this button, then change the value to 300, and then click OK.

e) Now you have created one range called Low Range. Repeat the above step to create one more range:

(1) High Range with start value 300 and end value 1000

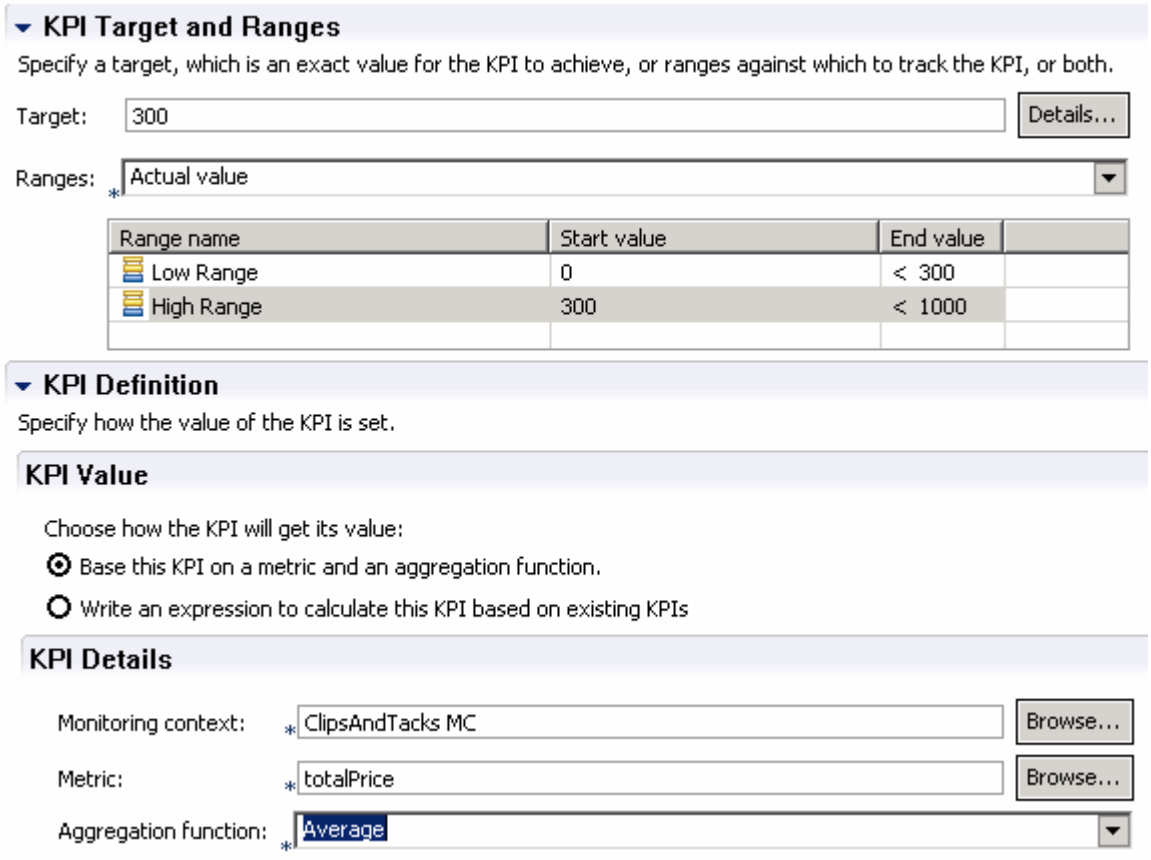
f) KPI Value – select '**Base this KPI on a metric and an aggregation function**'

g) Under KPI Details, for Monitoring context, click Browse…, then select '**ClipsAndTacks MC**', then click OK

h) Metric – click Browse…, then select **totalPrice**, then click OK

i) Aggregation function – select **Average**.

4) Press **Ctrl-S** to save your work.



____ 22. **(Optional)** To demonstrate the use of a counter, you can optionally use this section to create a counter which is incremented if you ship the order (rather than canceling the order). For every monitoring context, this counter is either zero or one depending on whether the order is cancelled or shipped. Counters would be more useful in other scenarios – perhaps to count the number of times an iterated activity is executed.

__ a. In the Monitor Details Model navigation view, right click on ClipsAndTacks MC, then select **New > Counter**. Enter these values:

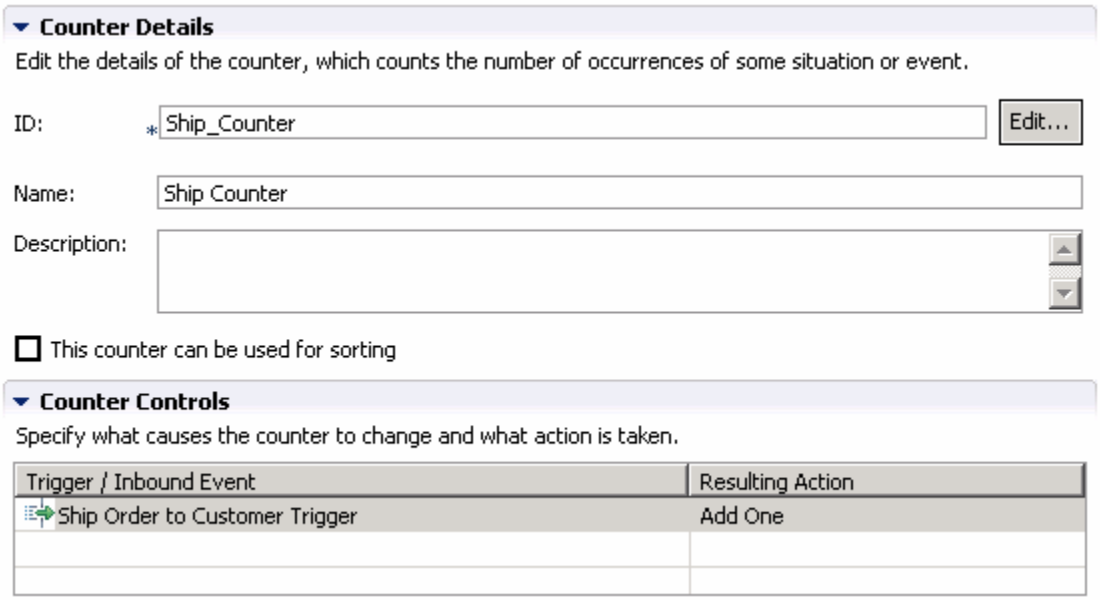
1) Name :- **Ship Counter**

2) ID :- **Ship_Counter**

3) For Counter Controls, click Add, then select **ClipsAndTacks MC → Ship Order to Customer Trigger**, click OK

4) Note that the Resulting Action defaults to **Add One**, which is correct.

__ b. Press **Ctrl-S** to save your work



____ 23. **(Optional))** To demonstrate the use of a stopwatch, you can optionally use this section to create a timer which will measure the duration of the monitoring context. You might find this useful in the dashboard instances view to quickly see which monitoring contexts have not been terminated.

__ a. In the model navigation view, right click on ClipsAndTacks MC, then select **New → Stopwatch**. Enter these values:

1) Name :- **Monitoring Context Timer**

2) ID :- **Monitoring_Context_Timer**

3) For Stopwatch Controls, click Add, then select **ClipsAndTacks MC → New Order Trigger** and then click OK.

4) Note that the default Resulting Action is **Start**

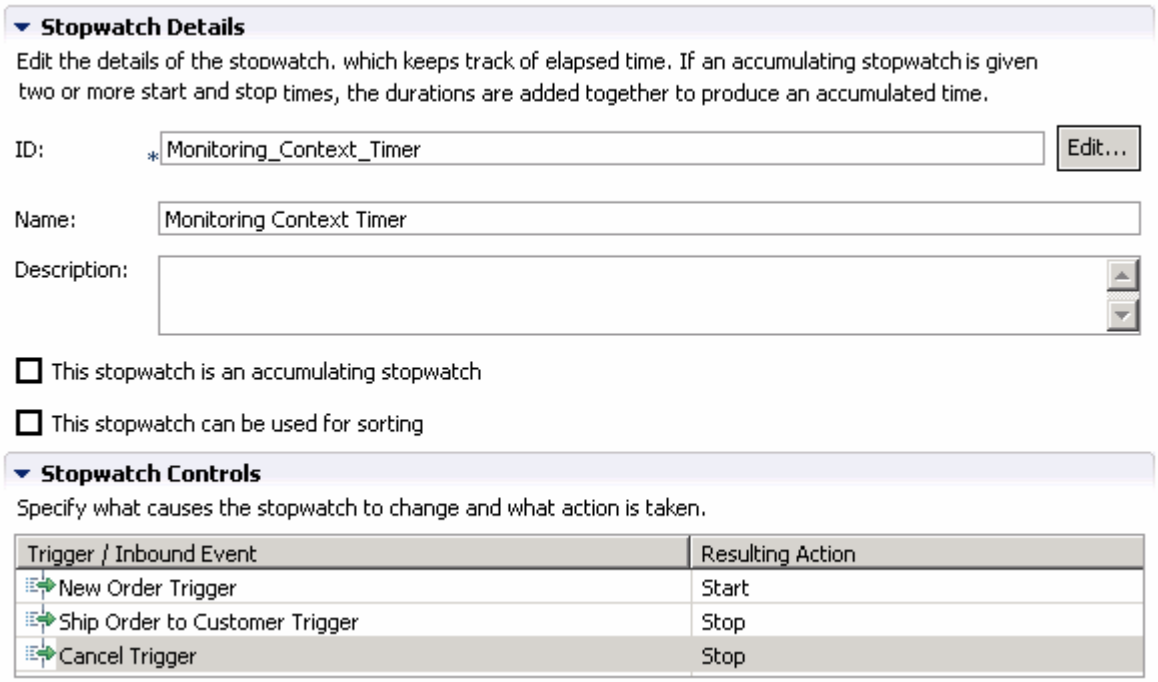
5) For Stopwatch Controls, click Add, then select **ClipsAndTacks MC → Ship Order to Customer Trigger** and then click OK.

6) Note that the default Resulting Action is Start, but you need to change it to **Stop** for this trigger. So click on the second row in column Resulting Action. A drop down box is shown, so pick **Stop** in the list.

7) For Stopwatch Controls, click Add, then select **ClipsAndTacks MC → Cancel Trigger** and then click OK.

8) Note that the default Resulting Action is Start, but you need to change it to **Stop** for this trigger. So click on the second row in column Resulting Action. A drop down box is shown, so pick **Stop** in the list.

__ b. Press **Ctrl-S** to save your work.



____ 24. Create a business situation event when the number of declined orders is greater than or equal to 3. First you will create an event definition for the outbound situation event. You will create a field called BusinessSituationName which is a required field for the outbound situation event. You will create a KPI that will track the total number of declined orders. Then you will create a trigger which is fired when the number of declined orders is greater than or equal to 3. Then you will define an outbound event which is fired by the trigger. In this outbound event, you will provide a value for the field BusinessSituationName which is used later when configuring action manager in the server administrative console. Create an new XSD event definition for declined orders

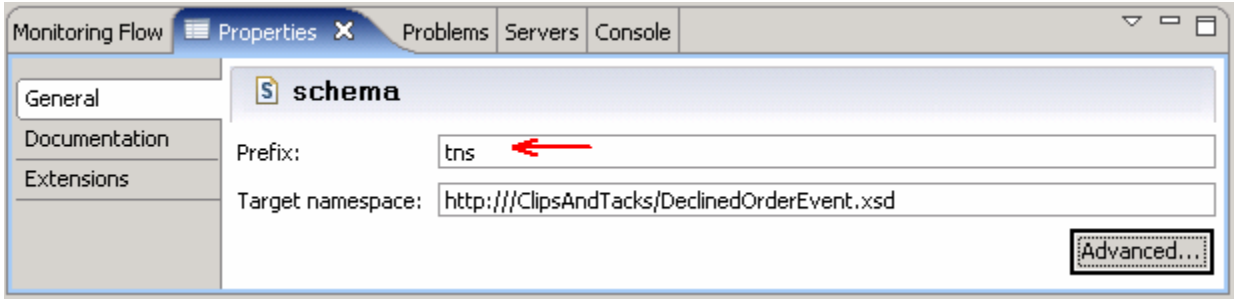
__ a. In the Business Monitoring project explorer view, expand **ClipsAndTacks**, right-click on **Event Definitions** and select **New → Event Definition… (xsd)** from the pop-up menu

__ b. In the **New XML Schema** panel, select **ClipsAndTacks** as the parent folder and enter the 'File name:' as **DeclinedOrderEvent.xsd** and click **Finish**

__ c. Click '**Yes**' in the question dialog if it appears

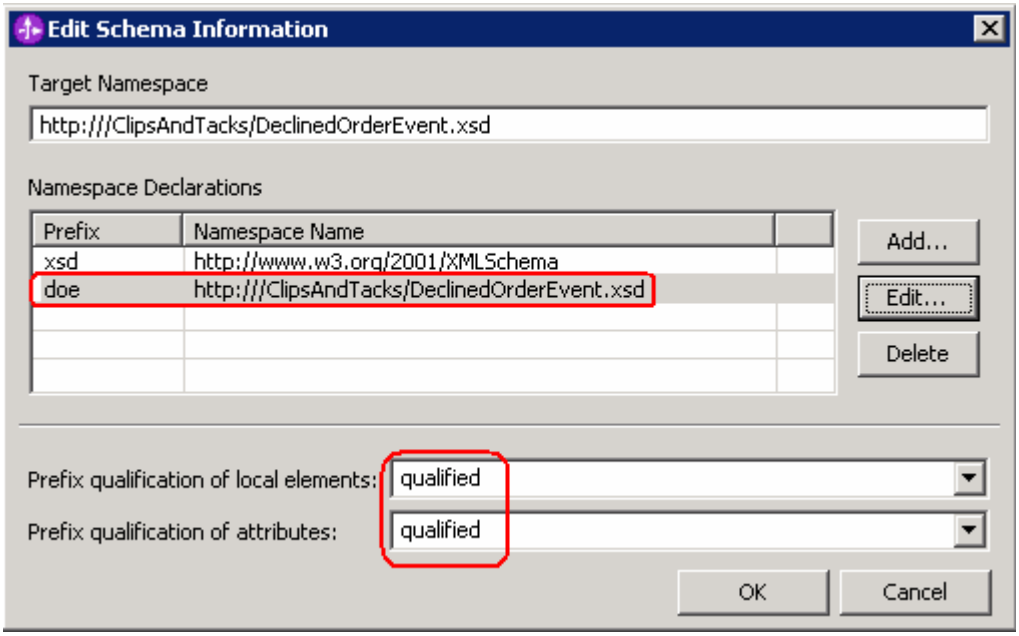
____ 25. Review and update the <schema> element, which is the root element of the schema

__ a. In the XSD editor, select the schema element and then select the '**General**' tab of the '**Properties**' view



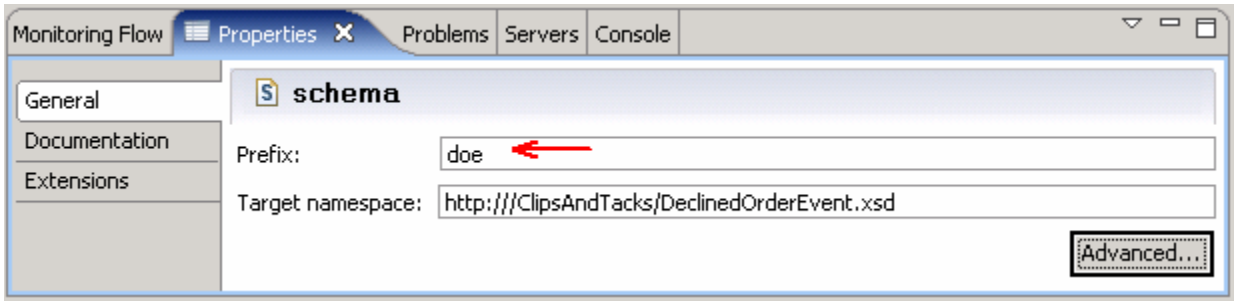
__ b. Update the '**Prefix:**' from **tns** to **doe** (where as **doe** stands for declined order event and the prefix resembles the name of the schema)

__ c. Click the **Advanced** button to edit the schema information (attributes)



__ d. Ensure that the prefix qualification of local elements and the attributes is set to '**qualified**'

__ e. Click **OK**. The '**General**' properties view of the 'ActivityEvent' schema should look like the picture below:

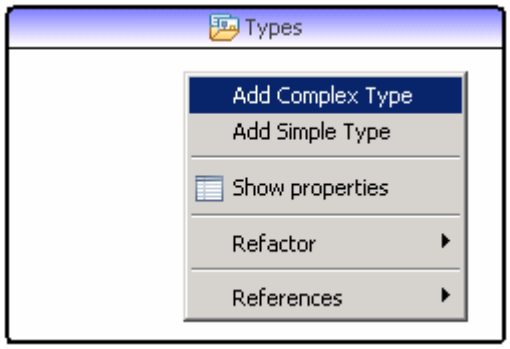


__ f. Save the changes. **File → Save** or **Ctrl + S**

____ 26. The 'DeclinedOrderEvent.xsd' schema opens in a '**Simplified**' XSD schema editor. Click on the pull-down button (▾) located on the top right corner of the XSD schema editor to select '**Detailed**' from the drop down list

____ 27. Add a complex element type named, **DeclinedOrderEvent**.

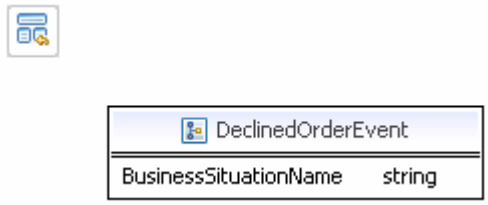
__ a. Right click in the **Types** section of the XSD editor and select '**Add Complex Type**' from the pop-up menu



__ b. Name the complex type as **DeclinedOrderEvent** and save the changes. You must see the new complex type element added as shown below:

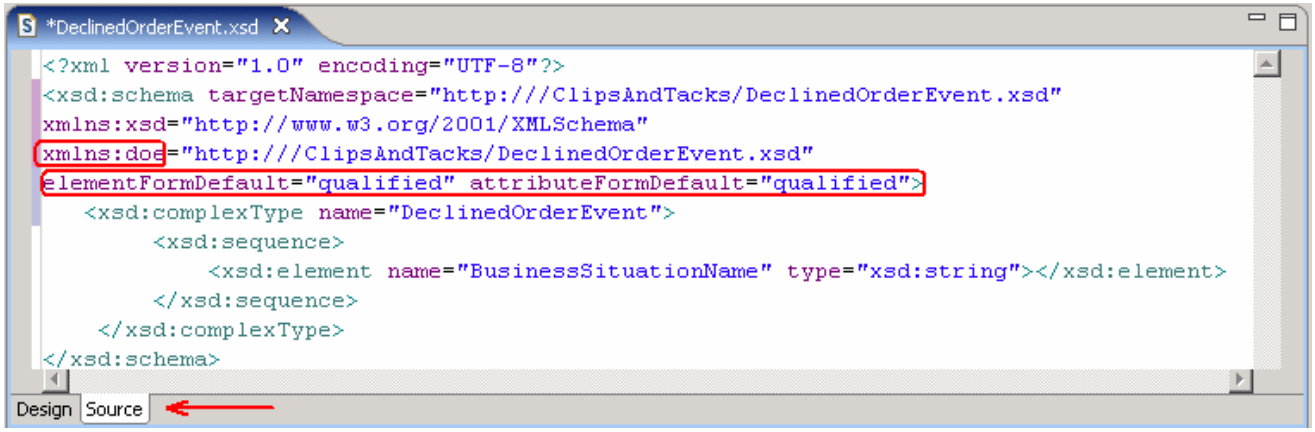
__ c. Now, right click the **DeclinedOrderEvent**, complex type from the **Types** section and select **Add Element** from the pop-up menu. An new element named 'NewElement' of type 'string' is created

__ d. Rename the element as '**BusinessSituationName**' and accept '**string**' as the type



__ e. Save the changes. **File → Save** or **Ctrl + S**.

__ f. The following is the source of the DeclinedOrderEvent schema you created:



____ 28. Create a KPI for declined orders.

1) Click the **KPI Model tab** for the ClipsAndTacks model.

2) In the navigation view of the KPI Model, right click on My KPI Context, then select **New → KPI**

   a) Name :- **Declined Order KPI**

   b) ID :- **Declined_Order_KPI**

3) Enter these values:

   a) Type :- **Decimal**

   b) In KPI Target and Ranges, for Target click **Details**…, then change the value to **3**, then click OK

   c) In KPI Target and Ranges, for Ranges select **Actual value**

   d) In the Range table,

      (1) Click Add

      (2) Change the Name to **Low Range**, and the ID to **Low_Range**

      (3) A row is created in the Range table with a range name Low Range.

      (4) For this row, select the cell in the Start value column and a button is displayed. Click this button, then change the value to 0, and then click OK.

      (5) For this row, select the cell in the End value column and a button is displayed. Click this button, then change the value to 3, and then click OK.

   e) Now you have created one range called Low Range. Repeat the above step to create one more range:

      (1) High Range with start value 3 and end value 10

   f) KPI Value – select **'Base this KPI on a metric and an aggregation function**'

   g) Under KPI Details, for Monitoring context, click Browse…, then select **'ClipsAndTacks MC**', then click OK

   h) Metric – click Browse…, then select '**ClipsAndTacks Key**', then click OK

   i) Aggregation function – select **Count**.

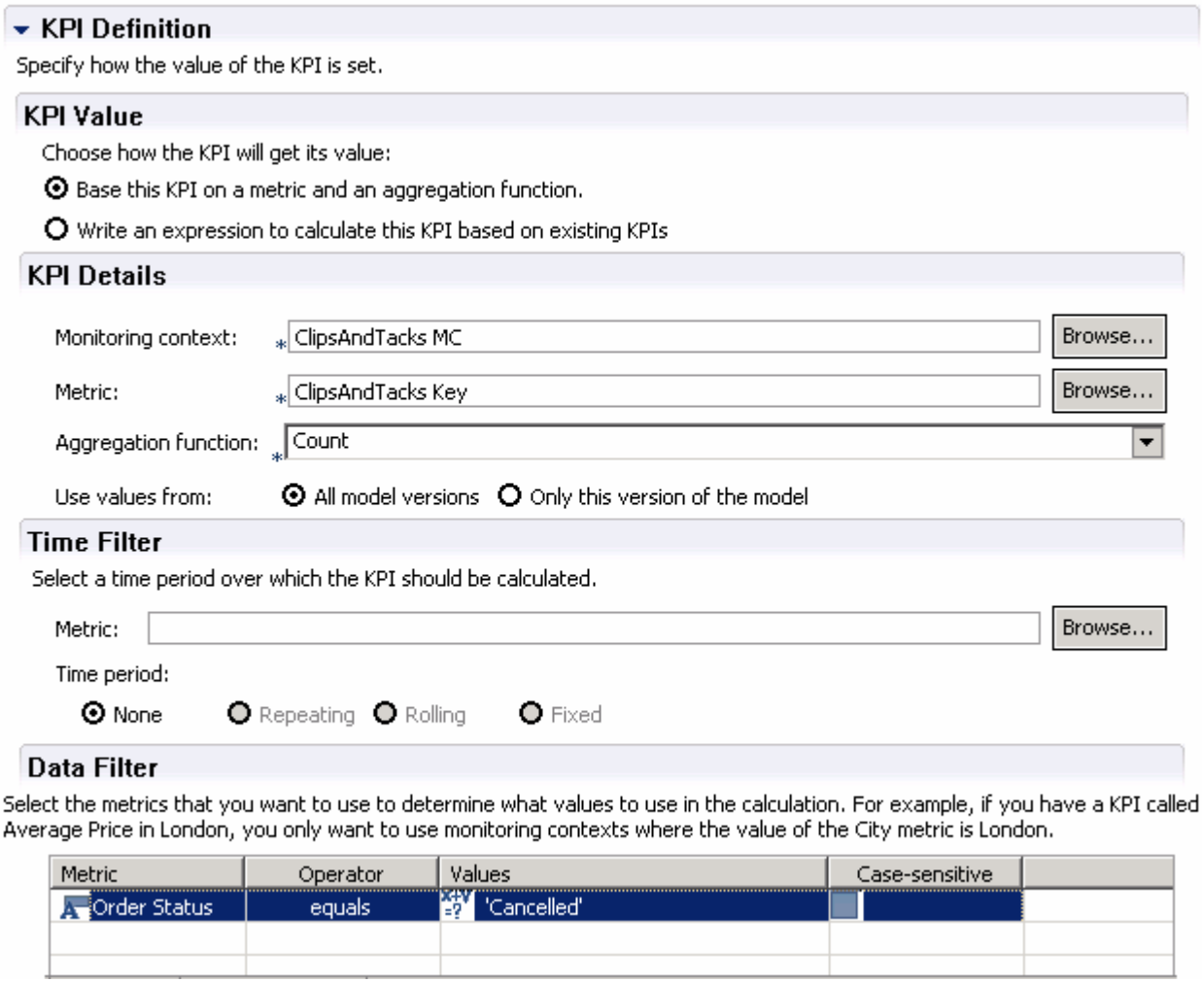
   j) In the Data Filter section for this KPI, click Add

   k) In the dialog, navigate to **ClipsAndTacks MC → Order Status**, then click OK

   l) In the row containing metric Order Status, make sure the operation defaults to 'equals'. Also, click the cell in the column Values, then enter this text (with quotation marks):

      'Cancelled'

4) Press **Ctrl-S** to save your work.



__ b. Create a trigger to fire the outbound situation event.

1) In the navigation view of the KPI Model, right click on My KPI Context, then select **New → Trigger**

   a) Name :- **Declined Order Trigger**

   b) ID :- **Declined_Order_Trigger**

2) Enter these values:

   a) Under Trigger Sources, click Add.

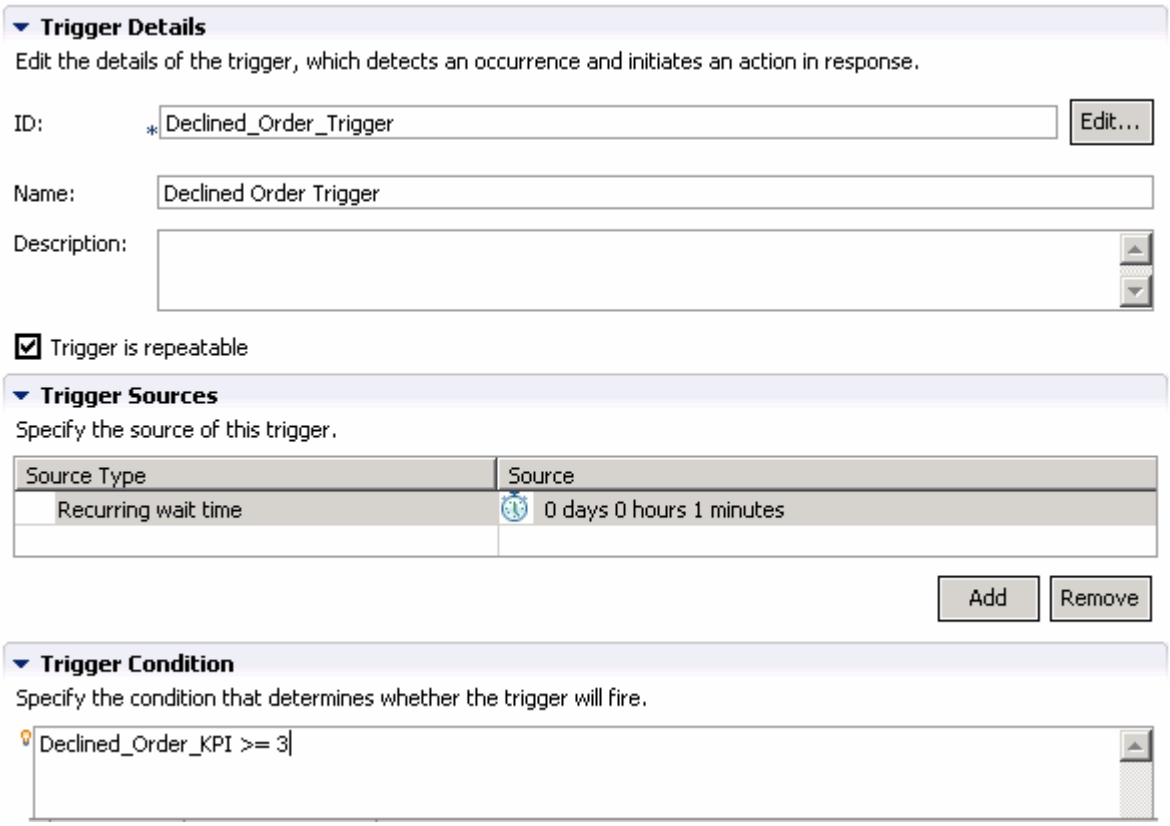
   b) Select **Recurring wait time**, then click OK

   c) In the Source column you see the default wait time is **1 minute**.

   d) In Trigger Condition, enter this text:

      Declined_Order_KPI >= 3

3) Press **Ctrl-S** to save your work.



__ c. Create an outbound situation event for declined orders.

1) In the navigation view of the KPI Model, right click on My KPI Context, then select **New > Outbound Event**

   a) Name :- **Declined Order Outbound Event**

   b) ID :- **Declined_Order_Outbound_Event**

   c) For trigger, click Browse…, then select **My KPI Context → Declined Order Trigger**, then click OK

2) Click the '**Add**' button for '**Event Parts**' to create a new event part type in the 'Event Type Details' section. The 'Create New Event part Type' panel opens

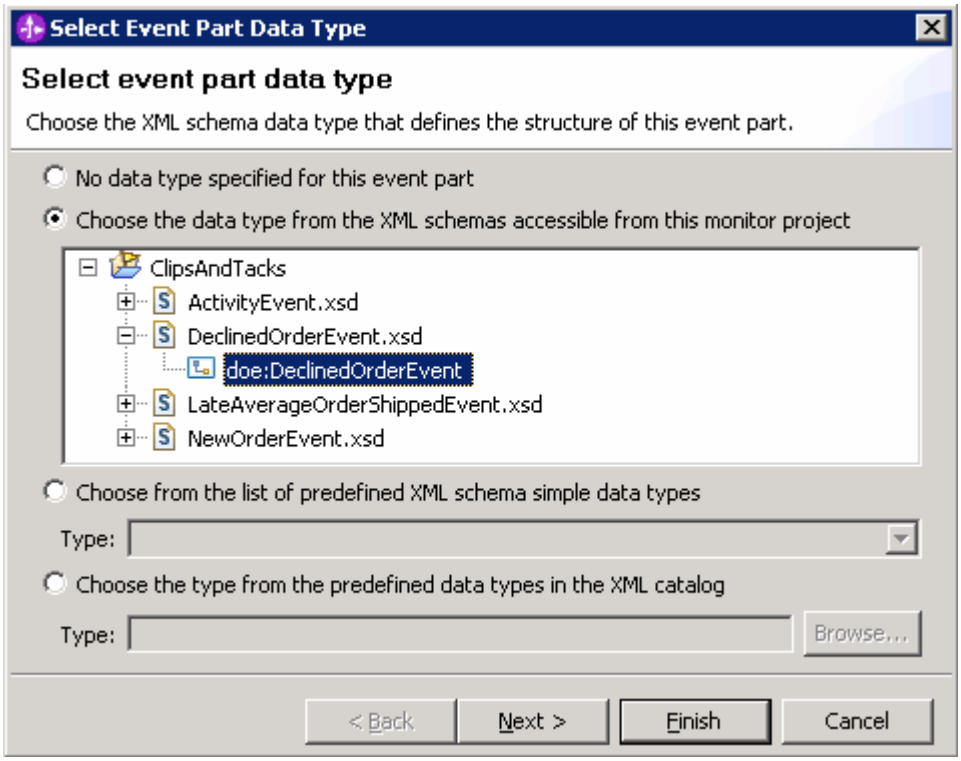
3) In the 'Create New Event part Type' panel, enter the following:

   a) Name : **My Event Part**

   b) ID : **My_Event_Part**

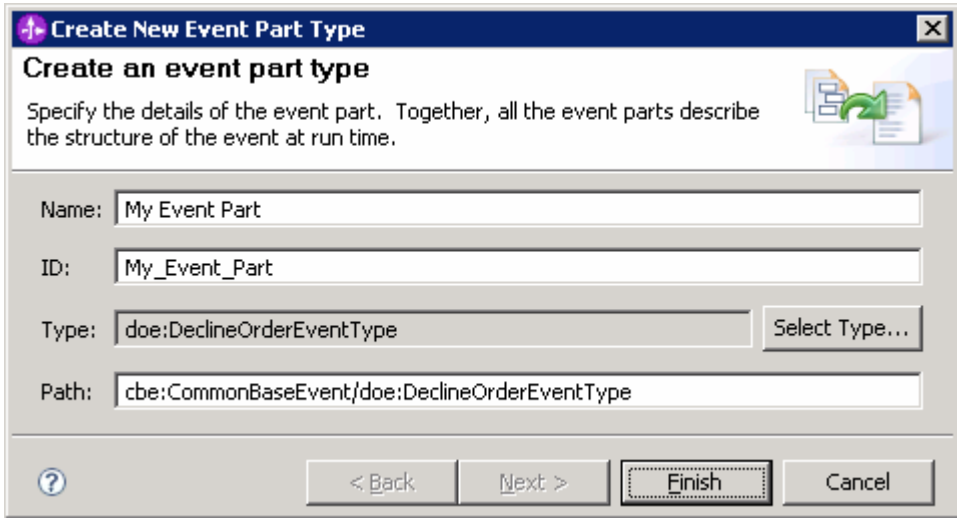
   c) Click the '**Select Type**' button for the 'Type' field. The 'Select Event Part Data Type' panel opens

   - Select the radio button for 'Choose the data type from XML accessible from this monitor project'
   - Expand **ClipsAndTacks → DeclinedOrderEvent.xsd** and select '**doe:DeclinedOrderEventType**'



- Click **Finish**

d) Path : **cbe:CommonBaseEvent/doe:DeclinedOrderEventType**

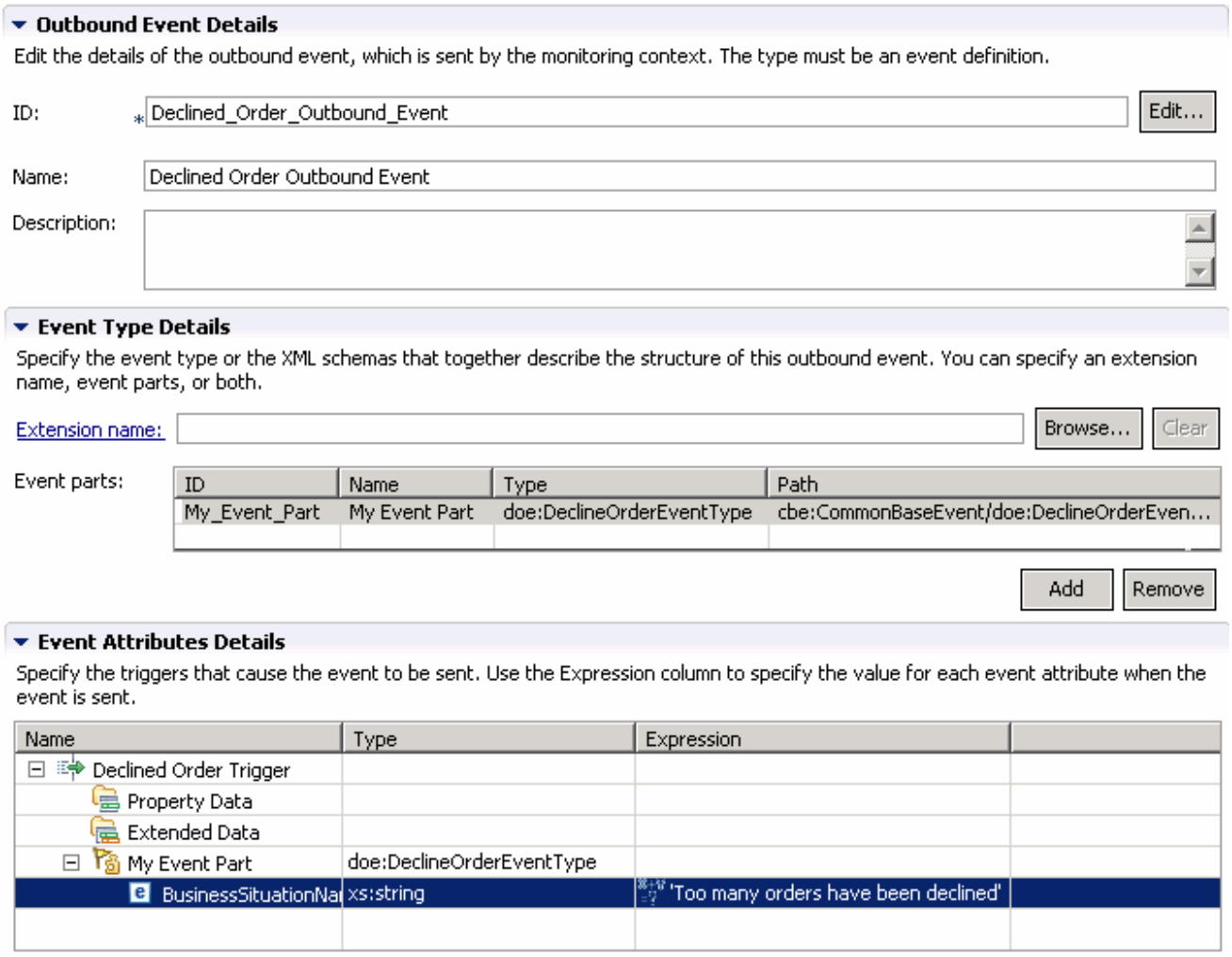


e) Click **Finish**

4) In the Event Attributes Details table, expand Declined Order Trigger → My Event Part, then on the same row as **BusinessSituationName**, set Expression to (include the quotation marks):

'Too many orders have been declined'

5) Press **Ctrl-S** to save your work.



____ 29. Create a business situation event when the order fulfillment duration is greater than 3 days. First you will create an event definition for the outbound situation event. You will create a field called BusinessSituationName which is a required field for the outbound situation event, and you will create another field for the average order processing time which will eventually be placed into the alert body. You do not need to create a KPI since there is already a KPI containing the average processing time. You will create a trigger which is fired when the average processing time is greater than 3 days. Then you will define an outbound event which is fired by the trigger. In this outbound event, you will provide a value for the field BusinessSituationName which is used later when configuring action manager in the server administrative console, and you will populate the field on the event containing the average order processing time. Create an event definition for average order processing time

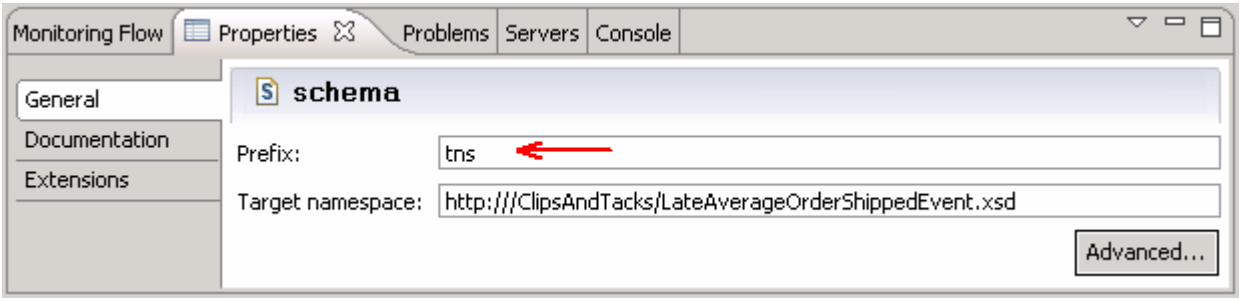
__ a. In the Business Monitoring project explorer view, expand **ClipsAndTacks**, right-click on **Event Definitions** and select **New → Event Definition… (xsd)** from the pop-up menu

__ b. In the **New XML Schema** panel, select **ClipsAndTacks** as the parent folder and enter the 'File name:' as **LateAverageOrderShippedEvent.xsd** and click **Finish**

__ c. Click '**Yes**' in the question dialog if it appears

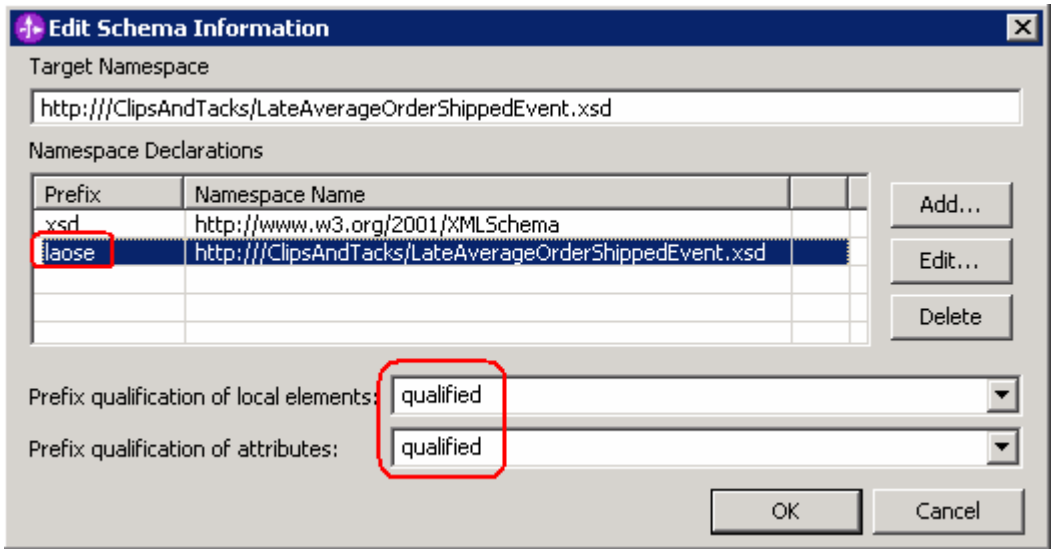
____ 30. Review and update the <schema> element, which is the root element of the schema

__ a. In the XSD editor, select the schema element and then select the '**General**' tab of the '**Properties**' view



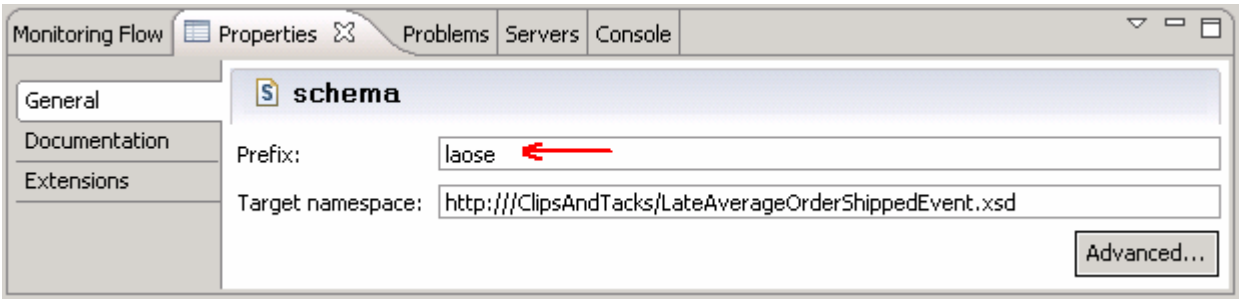
__ b. Update the '**Prefix:**' from **tns** to **laose** (where as **laose** stands for late average order shipped event and the prefix resembles the name of the schema)

__ c. Click the **Advanced** button to edit the schema information (attributes)



__ d. Ensure that the prefix qualification of local elements and the attributes is set to '**qualified**'

__ e. Click **OK**. The '**General**' properties view of the 'ActivityEvent' schema should look like the picture below:

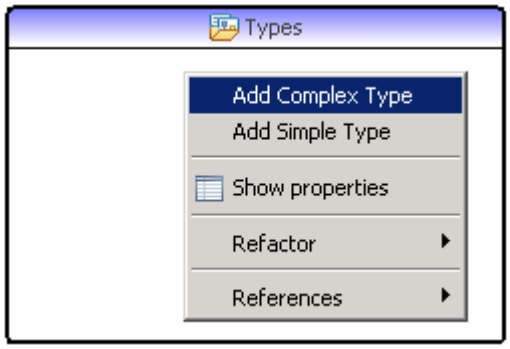


__ f. Save the changes. **File → Save** or **Ctrl + S**

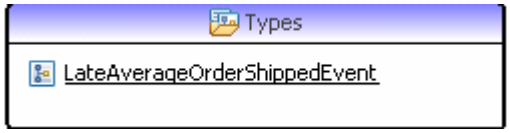
____ 31. The 'DeclinedOrderEvent.xsd' schema opens in a '**Simplified**' XSD schema editor. Click on the pull-down button (▾) located on the top right corner of the XSD schema editor to select '**Detailed**' from the drop down list

____ 32. Add a complex element type named, **LateAverageOrderShippedEvent**

__ a. Right click in the **Types** section of the XSD editor and select '**Add Complex Type**' from the pop-up menu

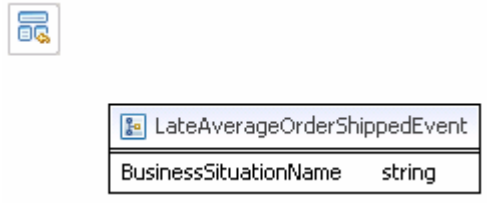


__ b. Name the complex type as **LateAverageOrderShippedEvent** and save the changes. You should see the new complex type element added as shown below:



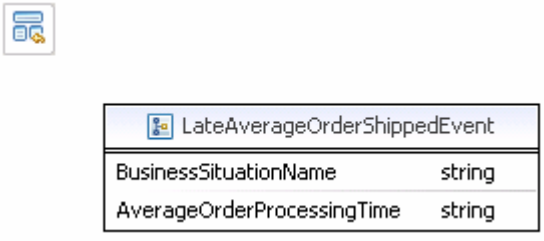
__ c. Now, right click the **LateAverageOrderShippedEvent**, complex type from the **Types** section and select **Add Element** from the pop-up menu. An new element named 'NewElement' of type 'string' is created

__ d. Rename the element as '**BusinessSituationName**' and accept '**string**' as the type



__ e. Right click the **LateAverageOrderShippedEvent**, complex type again from and select **Add Element** from the pop-up menu. An new element named 'NewElement' of type 'string' is created

__ f. Rename the element as '**AverageOrderProcessingTime**' and accept '**string**' as the type



__ g. Save the changes. **File → Save** or **Ctrl + S**.

__ h. The following is the source of the DeclinedOrderEvent schema you created:

____ 33. Create a trigger to fire the outbound situation event.

__ a. In the navigation view of the KPI Model, right click on My KPI Context, then select **New → Trigger**

1) Name :- **Order Fulfillment Timer Trigger**

2) ID :- **Order_Fulfillment_Timer_Trigger**

__ b. Enter these values:

a) Under Trigger Sources, click Add.

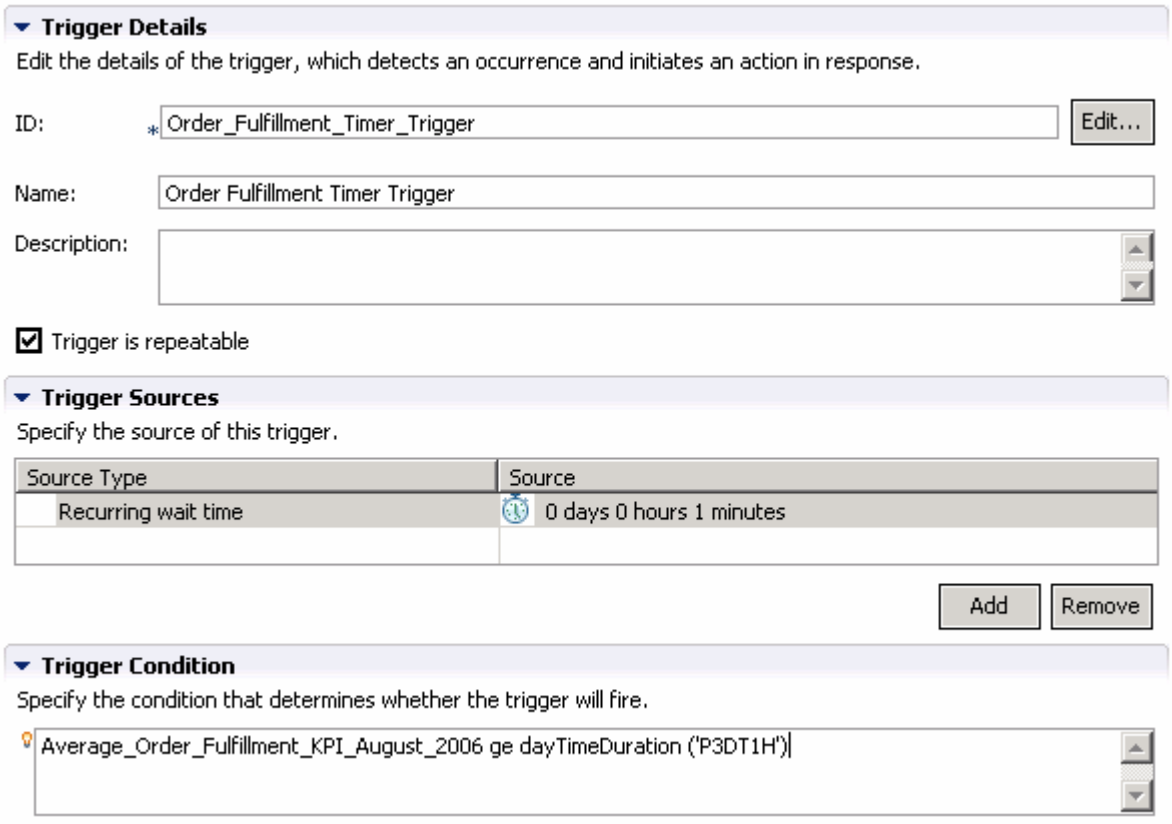
b) Select **Recurring wait time**, then click OK

c) In the Source column you see the default wait time is **1 minute**.

d) In Trigger Condition, enter this text:

Average_Order_Fulfillment_KPI_August_2006 ge xs:dayTimeDuration ('P3DT1H')

__ c. Press **Ctrl-S** to save your work.



__ d. Create an outbound situation event for average order processing time.

1) In the navigation view of the KPI Model, right click on My KPI Context, then select **New > Outbound Event**

   a) Name :- **Order Fulfillment Outbound Event**

   b) ID :- **Order_Fulfillment_Outbound_Event**

   c) For trigger, click Browse…, then select **My KPI Context → Order Fulfillment Timer Trigger**, then click OK

2) Click the '**Add**' button for '**Event Parts**' to create a new event part type in the 'Event Type Details' section. The 'Create New Event part Type' panel opens

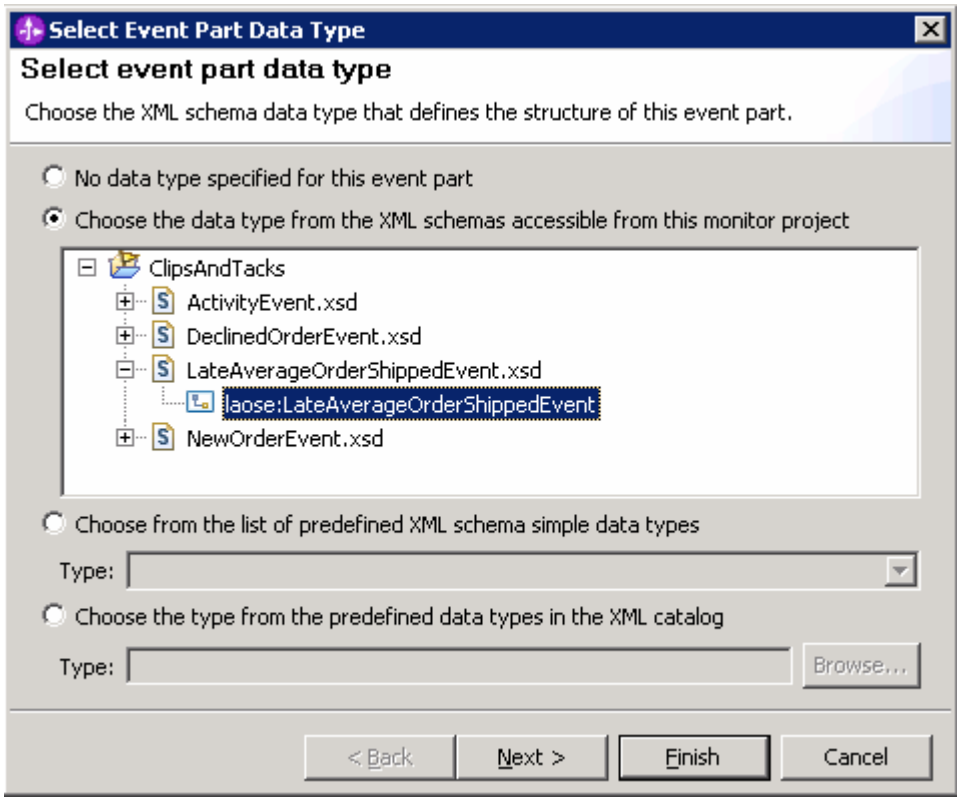
3) In the 'Create New Event part Type' panel, enter the following:

   a) Name : **My Event Part**

   b) ID : **My_Event_Part**

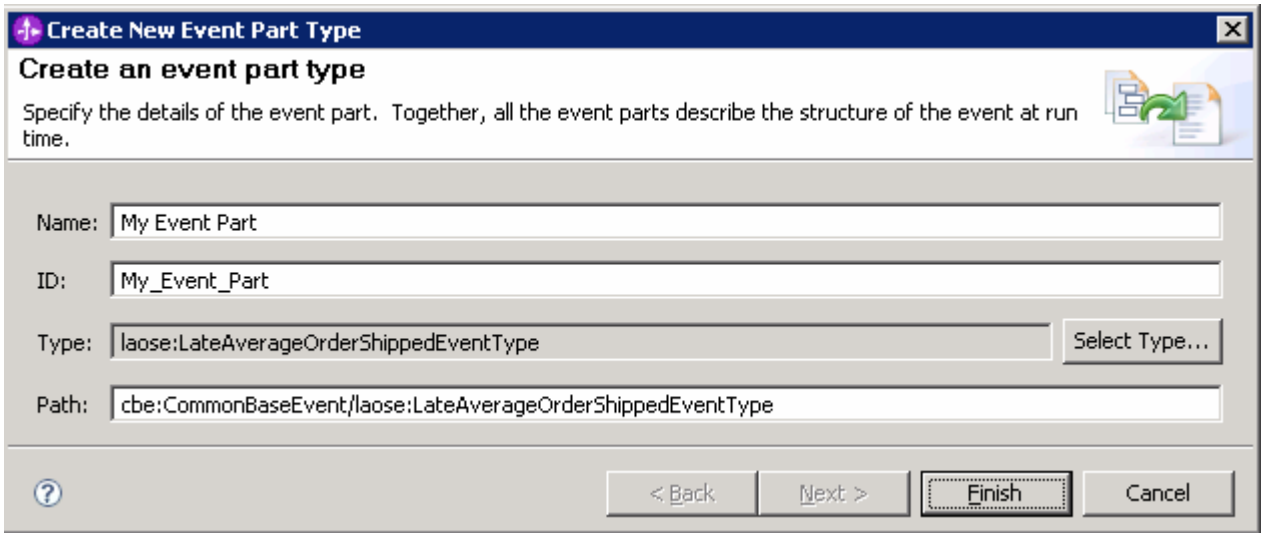
   c) Click the '**Select Type**' button for the 'Type' field. The 'Select Event Part Data Type' panel opens

   - Select the radio button for 'Choose the data type from XML accessible from this monitor project'
   - Expand **ClipsAndTacks → LateAverageOrderShippedEvent.xsd** and select '**laose: LateAverageOrderShippedEventType**'



- Click **Finish**

d) Path : **cbe:CommonBaseEvent/laose:LateAverageOrderShippedEventType**



e) Click **Finish**

4) In the Event Attributes Details table, expand Order Fulfillment Timer Trigger → My Event Part and then on the same row as **BusinessSituationName**, set Expression to (include the quotation marks):

‘Average shipment is too late’

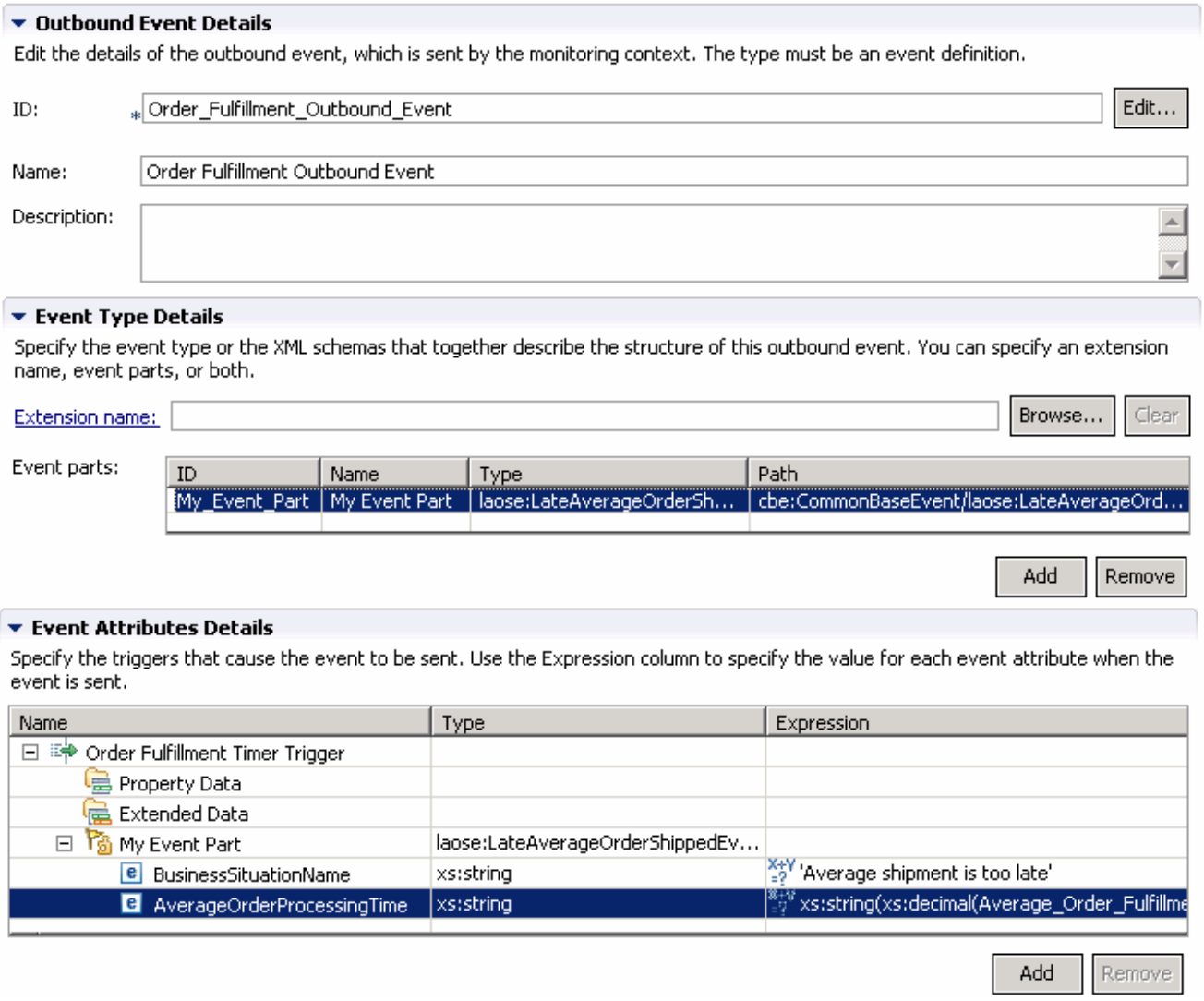
5) In the Event Attributes Details table, expand Order Fulfillment Timer Trigger → My Event Part, then on the same row as **AverageOrderProcessingTime**, set Expression to (include the quotation marks):

xs:string(xs:decimal(Average_Order_Fulfillment_KPI_August_2006) div 86400)

Note: Here the decimal function converts the duration to seconds, so then it can be divided by 86400 which will give you the number of days.

---

6) Press **Ctrl-S** to save your work

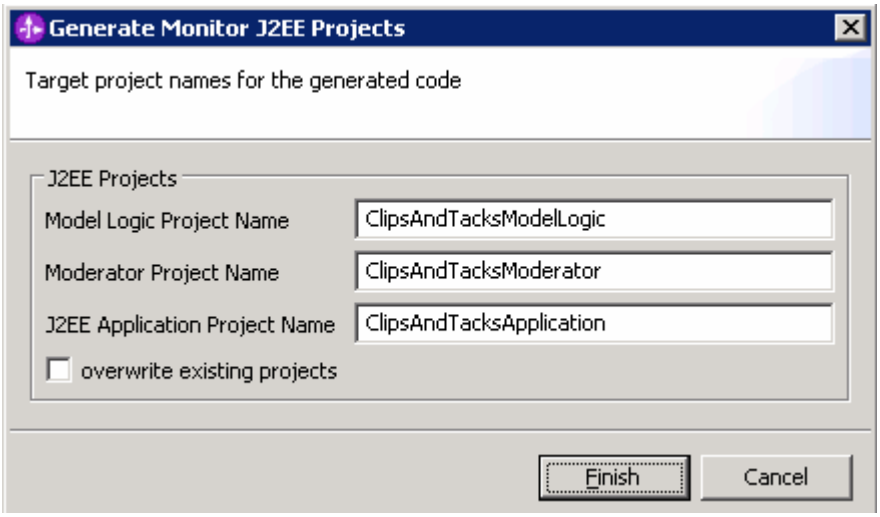


____ 34. Check for any errors in the Problems view. You should resolve any errors before continuing. Warnings and informational messages may be present but these are not a problem.

Part 2: Publish the model to the server

In this section you will use Rational Application Developer or WebSphere Integration Developer to publish the monitor model to the monitor server.

____ 1. In Project Explorer, expand **ClipsAndTacks → Monitor models → ClipsAndTacks.mm**. Right click over **ClipsAndTacks.mm** and then select **Generate Monitor J2EE Projects** from the pop-up menu



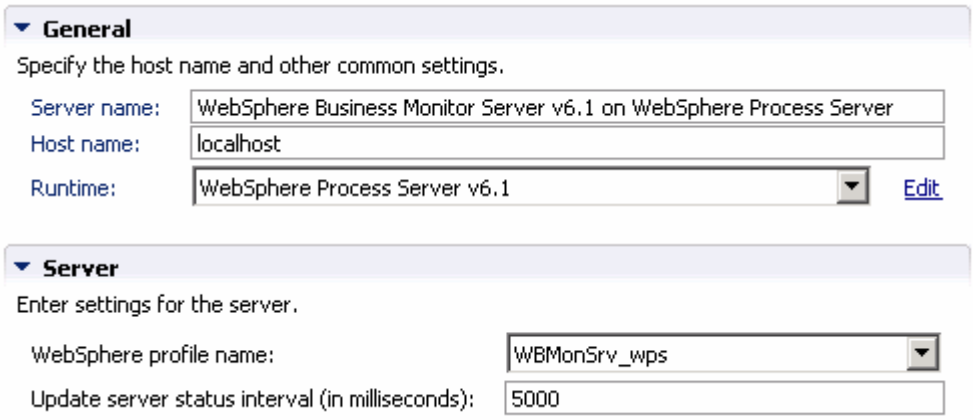
____ 2. A progress dialog shows the status of the operation and it closes when the operation is complete. Check for errors in the Problems view. There may be warnings, but there should not be any errors.

____ 3. If you are using the Monitor Toolkit, ensure that the correct Monitor server profile is setup in the Servers view.

__ a. Click the Servers tab.

__ b. Double click the listed server to open the server editor.

__ c. If you are using Rational Application Developer, ensure **WBMonSrv** is selected for the WebSphere profile name. If you are using WebSphere Integration Developer, ensure **WBMonSrv_wps** is selected for the WebSphere profile name.

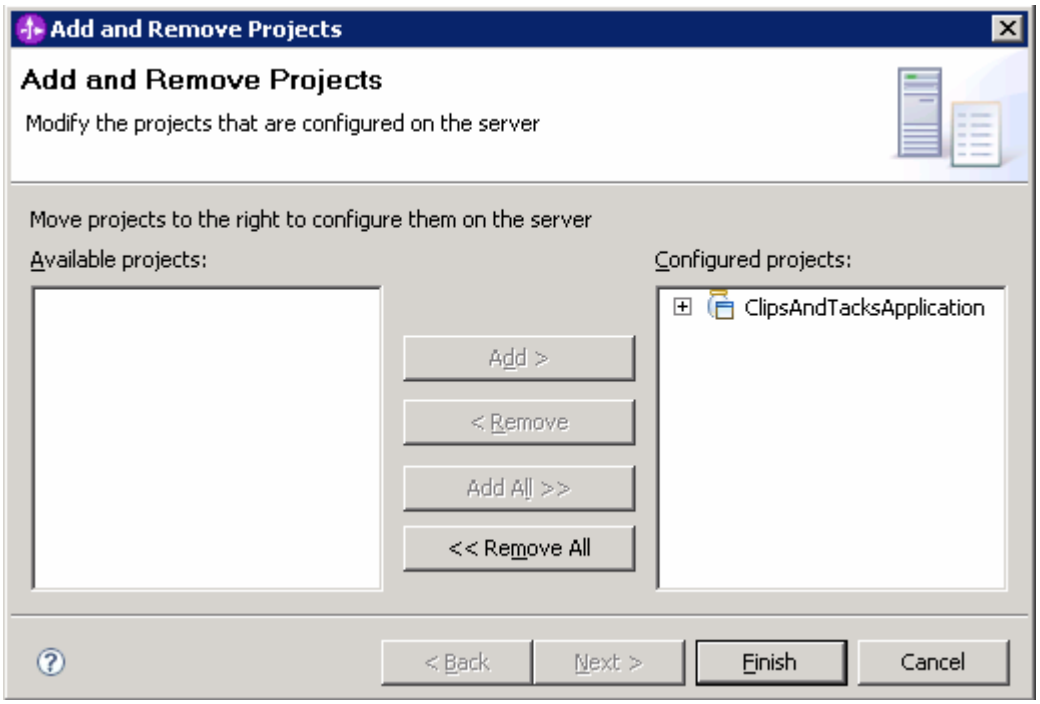


__ d. Press **Ctrl-S** to save your work.

____ 4. Click the Servers tab, then right click and select the **Start** option to start the server. This may take a few minutes to complete

____ 5. Right click in the servers view, then select **Add and Remove Projects….**

____ 6. Click **Add** to move the ClipsAndTacksApplication from the list of available projects to the list of configured projects.



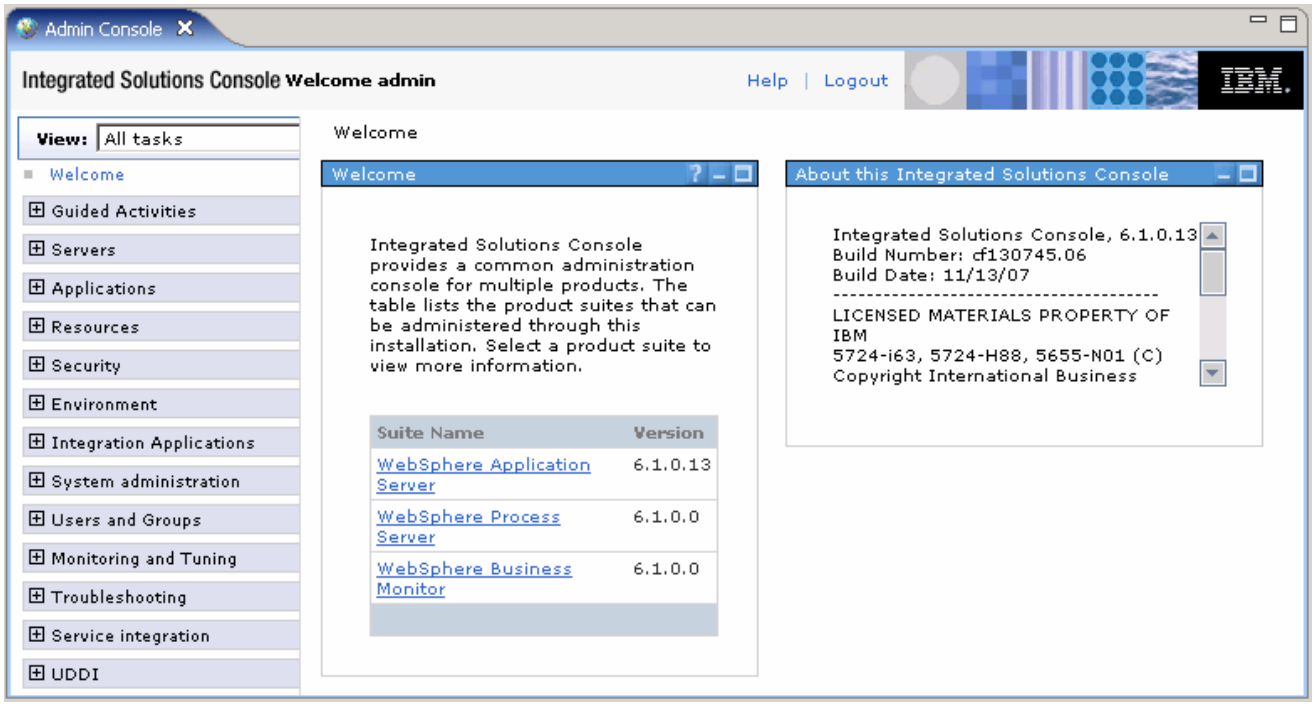
____ 7. Click **Finish**.

____ 8. A progress message is displayed in the lower right corner of the window

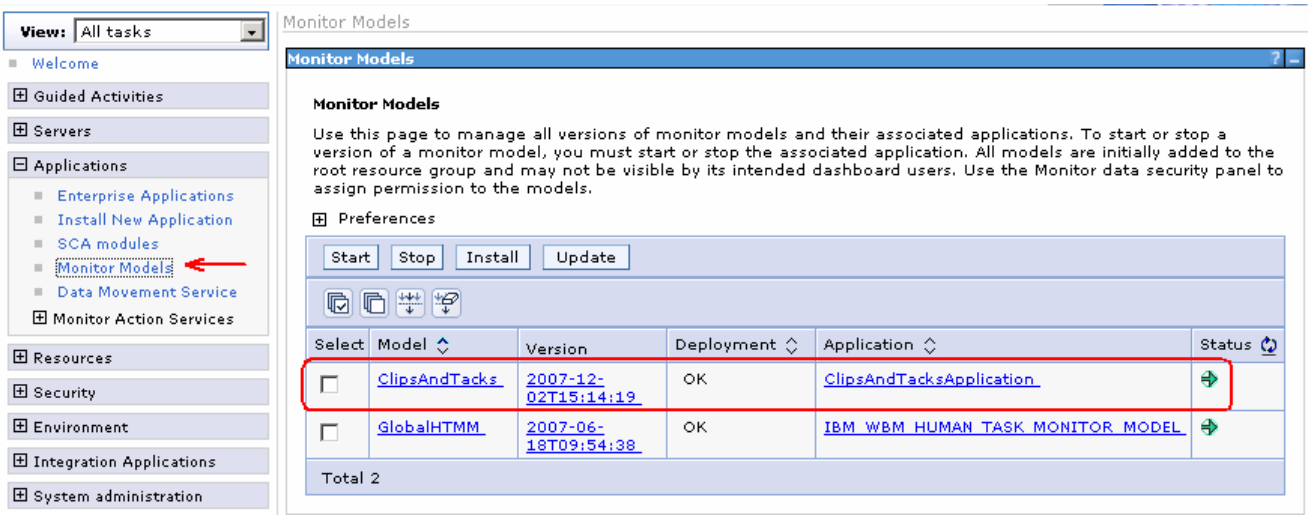
____ 9. Check the messages in the console view. You should see the following message when the application has been started:

Application started: ClipsAndTacksApplication

____ 10. In the servers view, right click, then select **Run administrative console**. Enter the user name as **admin** and password as **admin** when prompted. You should see it open in a separate tab:



____ 11. Click **Applications > Monitor Models**. The application should show green status if it started successfully.



____ 12. If the model shows red (stopped), then wait a moment, then refresh by clicking on the icon to the right of Status in the last column of the table. You should see green (started) for the model. If it does not show green, be patient and keep refreshing until it does show green.

____ 13. Check the server log to ensure there are no problems. You can check this in the console view.

Part 3: Configure action manager for business situation events

This section shows you how to configure action manager to create alerts based on business situation events defined in the monitor model. The alerts will be visible in the Alert view on your dashboard.

You do not need an LDAP server to test the situation events.

____ 1. Add a template for declined order alerts.

__ a. In the administrative console, navigate to **Applications > Monitor Action Services > Template Definitions > Notifications**

__ b. Click **New.**

__ c. Enter **AlertDeclined** as Template name and a description**.**

__ d. Select **Dashboard Alert.**

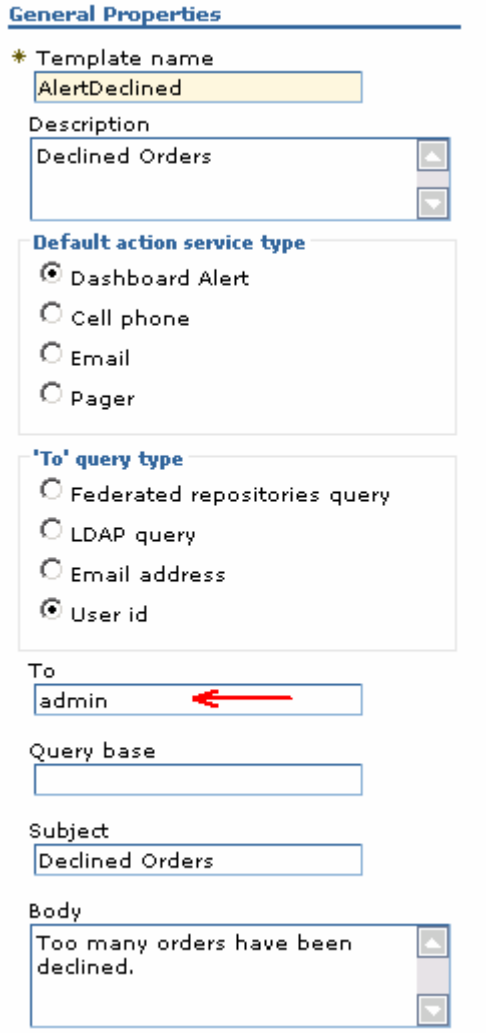
__ e. Select **User id.**

__ f. For the **To** field, enter **admin.** This 'User id' is the one that will receive the alert, so you should log into the dashboard with this 'User id'.

Note: The user **admin** is the default username configured for Monitor Test Environment profile

__ g. Enter a subject.

__ h. Enter the body.



__ i. Click **OK**.

____ 2. Add a template for order processing time alerts.

__ a. In the administrative console, navigate to **Applications → Monitor Action Services → Template Definitions → Notifications**

__ b. Click **New.**

__ c. Enter **AlertLate** as Template name and a description**.**

__ d. Select **Dashboard Alert**

__ e. Select **User id**

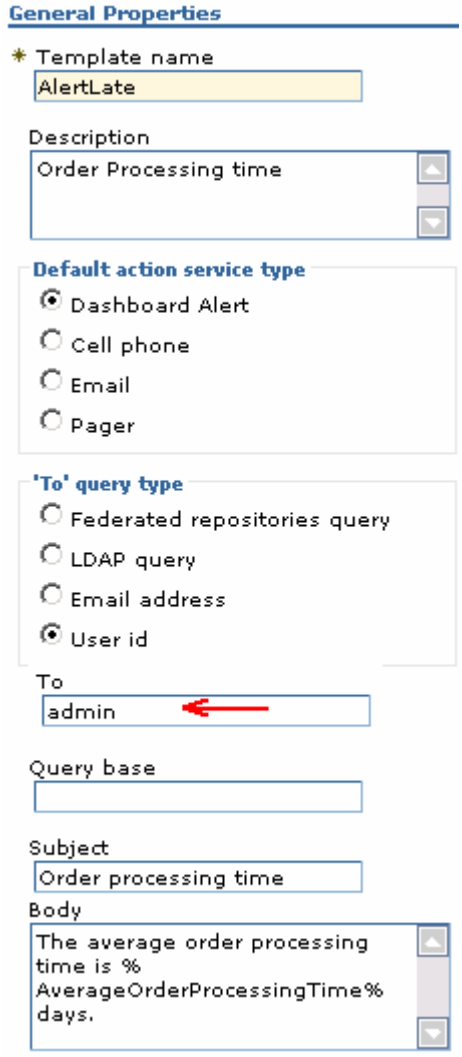
__ f. For the **To** field, enter **admin.** This 'User id' is the one that will receive the alert, so you should log into the dashboard with this 'User id'.

Note: The user **admin** is the default username configured for Monitor Test Environment profile

__ g. Enter a subject.

__ h. Enter the body, and you may optionally enter a substitution variable as shown below. For the actual alert that is sent, the average order processing time from the event payload will be substituted for %AverageOrderProcessingTime% in the body. For example:

The average order processing time is %AverageOrderProcessingTime% days.



__ i. Click **OK**.

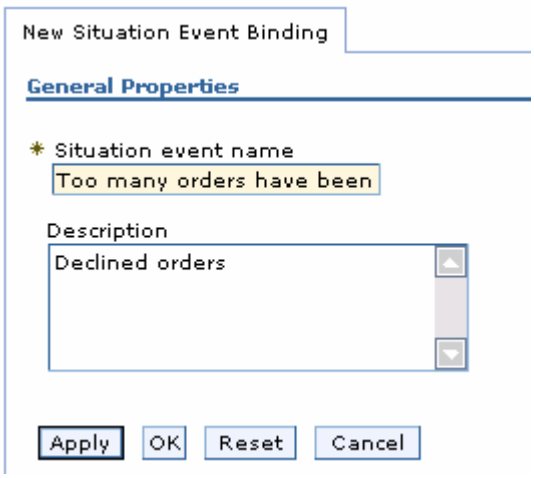
____ 3. Add the binding from the situation event to the action type for declined orders.

__ a. In the administrative console, navigate to **Applications → Monitor Action Services → Installed Situation Event Bindings**

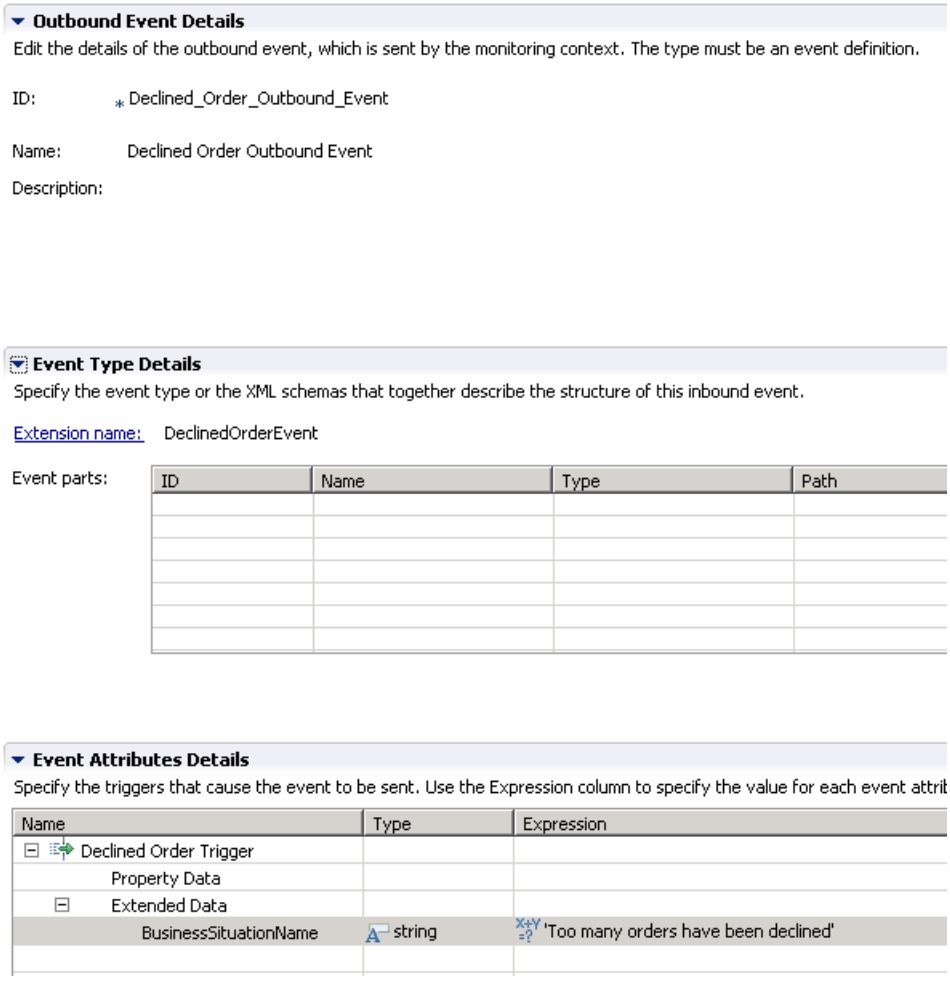
__ b. Click **New**.

__ c. Enter the situation event name that you defined in the model. In this lab, you created a business situation name as follows:

Too many orders have been declined



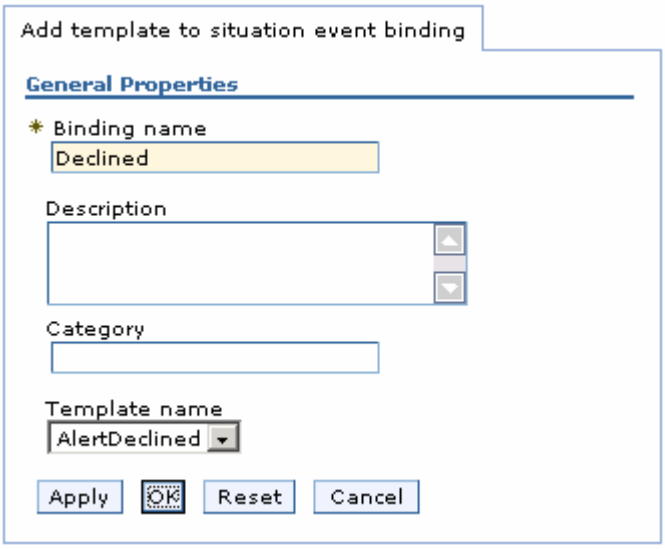
__ d. This should match exactly the value in the BusinessSituationName field in the outbound event. You may want to copy/paste from the model. Here is a screen capture from the model showing the Declined Order Outbound Event:



__ e. Enter a description, then click **Apply**.

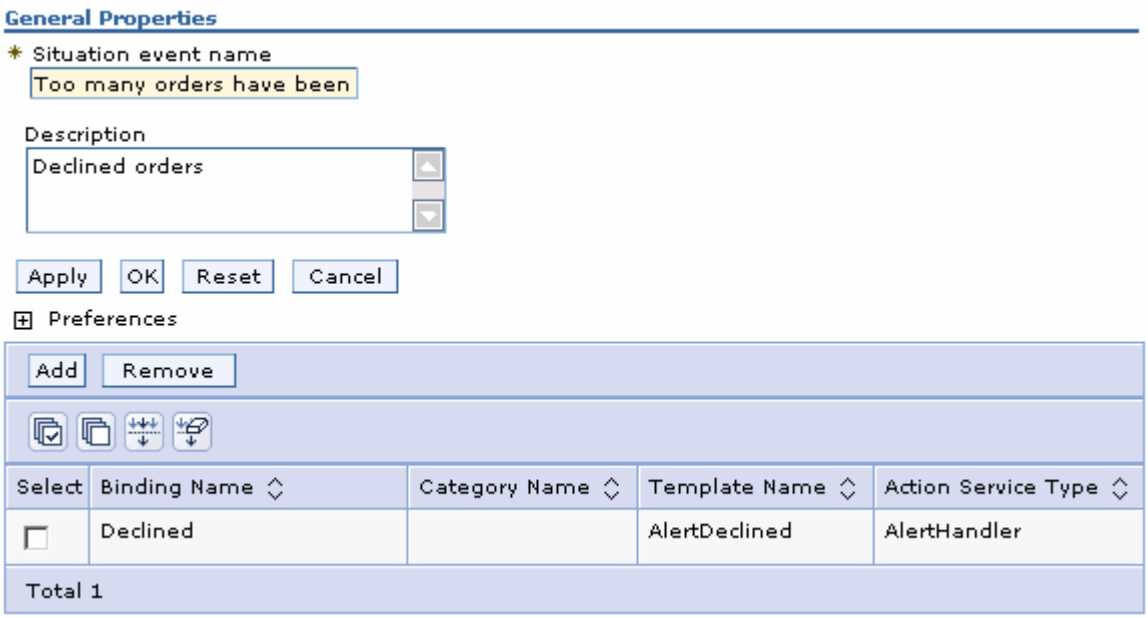
__ f. Click **Add**.

__ g. Enter a binding name, then select the template **AlertDeclined**



__ h. Click **OK**.

__ i. Notice that you now have one action defined for this situation event. If you had other action templates defined then you could add more actions to this event. So then you could send a notification for this situation to multiple destinations including e-mail, alerts and Web services.



__ j. Click **OK**.

____ 4. Add the binding from the situation event to the action type for order processing time.

__ a. In the administrative console, navigate to **Applications > Monitor Action Services > Installed Situation Event Bindings**

__ b. Click **New**.

__ c. Enter the situation event name that you defined in the model. In this lab, you created a business situation name as follows:

Average shipment is too late

__ d. . This should match exactly the value in the BusinessSituationName field in the outbound event. You may want to copy/paste from the model

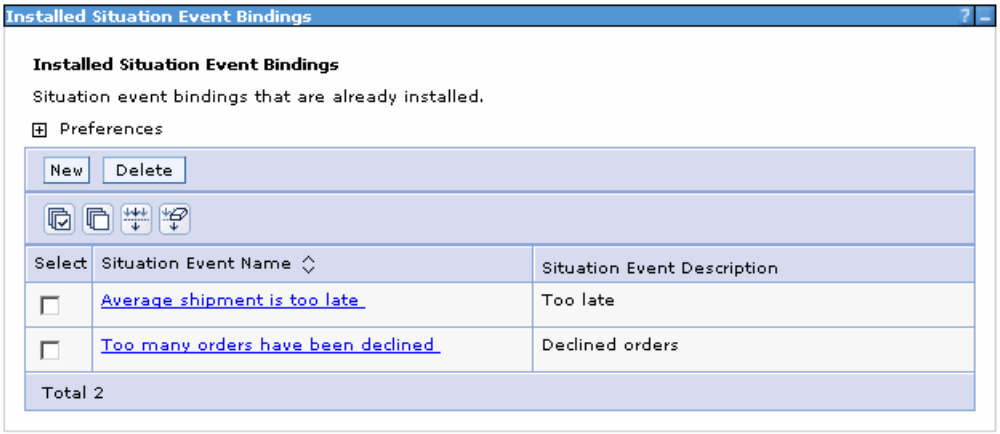
__ e. Enter a description, then click **Apply**.

__ f. Click **Add**.

__ g. Enter a binding name, then select the template **AlertLate**, then click **OK**.

__ h. Click **OK**.

____ 5. In the administrative console, navigate to **Applications → Monitor Action Services → Installed Situation Event Bindings.** You should see two bindings as follows:



Part 4: Run events to exercise the model

Rather than installing a J2EE application to actually create the events that you want to monitor, you are going to use a program to simulate the submission of events from the application. The supplied program is 'BatchCBEWriter61' and it will submit the events to the Common Event Infrastructure. Look for the program in \Labfiles61\ClipsAndTacks\BatchCBEWriter. This program reads XML files which represent the Common Base Events for the model.

____ 1. Update BatchCBEWriter61.bat to point WAS_HOME to the monitor server home, for example '**set WAS_HOME=C:\IBM\WebSphere\MonServer**'. If you are using the integrated monitor server in WebSphere Integration Developer, the path would be **<WID_HOME>\runtimes\bi_v61**. If you are using the integrated monitor server in Rational Application Developer, the path would be **<RAD_HOME>\runtimes\base_v61**.

____ 2. Update **config.properties**, setting the **serverName** and **portNumber**. You can find the port number by browsing the server log and finding '**bootstrap port**'. For example, check for the log at C:\IBM\WebSphere\MonServer\profiles\WBMon01\logs\server1\SystemOut.log. For the integrated server in WebSphere Integration Developer, the path would be <WID_HOME>\pf\WBMonSrv_wps\logs\server1\SystemOut.log and the portNumber should be 2810. For the integrated server in Rational Application Developer, the path would be <RAD_HOME>\runtimes\base_v61\profiles\WBMonSrv\logs\server1\SystemOut.log and the portNumber should be 2810. Here is an example of the config.properties settings:

__ a. `connect.serverName = localhost`

__ b. `connect.portNumber = 2810`

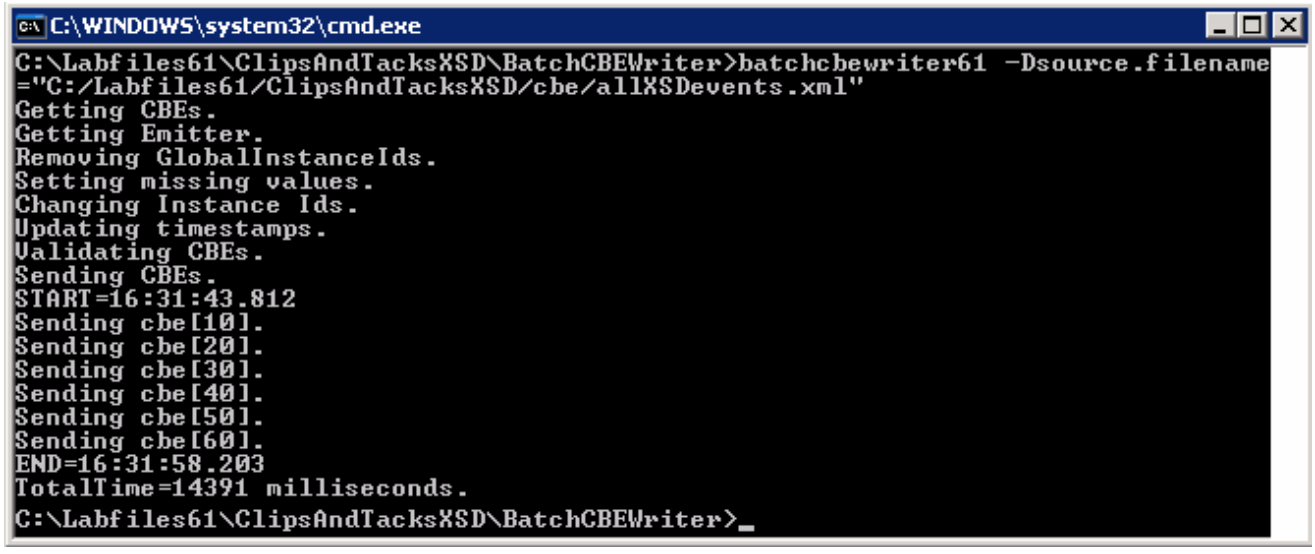
____ 3. Open a command window, then change directory to the folder containing BatchCBEWriter61, for example, type this command

__ a. cd \Labfiles61\ClipsAndTacksXSD\BatchCBEWriter

____ 4. Run commands to load the common base events to the server.

__ a. batchcbewriter61 -Dsource.filename="C:/Labfiles61/ClipsAndTacksXSD/cbe/allXSDevents.xml"

____ 5. When you run BatchCBEWriter61, you should see results such as:



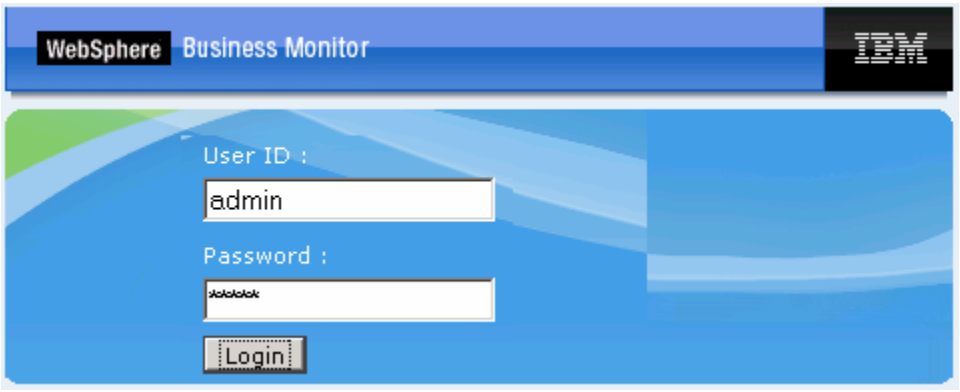
Part 5: Create a dashboard

In this section you will build a dashboard. You will add views to the dashboard and configure them.

____ 1. Open the dashboard.

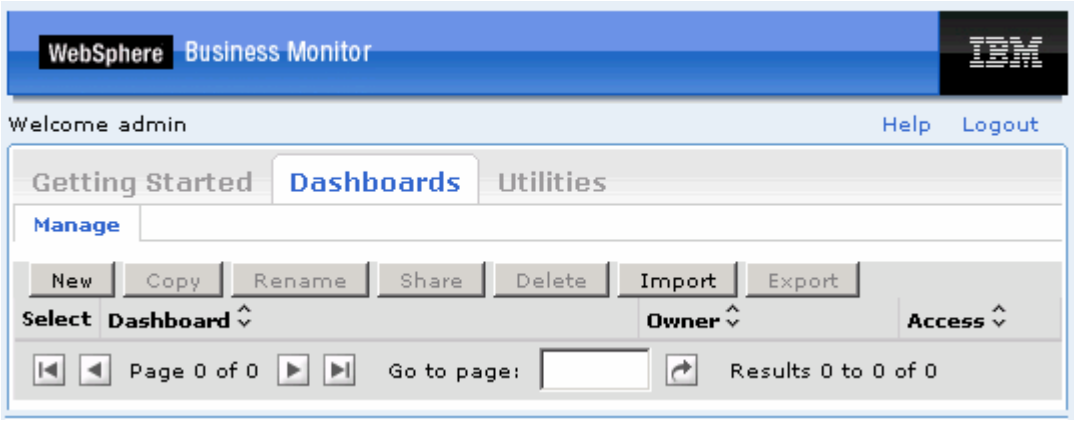
__ a. In Rational Application Developer or WebSphere Integration Developer, in the servers view, right click and select **WebSphere Business Monitor Dashboard**

__ b. The dashboard should open as follows:

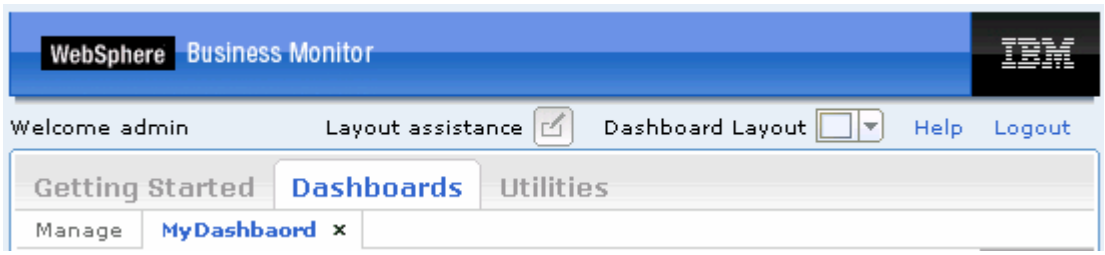


__ c. Enter **admin** as the username and **admin** as password and then click **Login**. You should use '**admin**' so that you can view the alerts which were setup in action manager to be sent to this particular 'User id'.

__ d. Click the **Dashboards** tab.

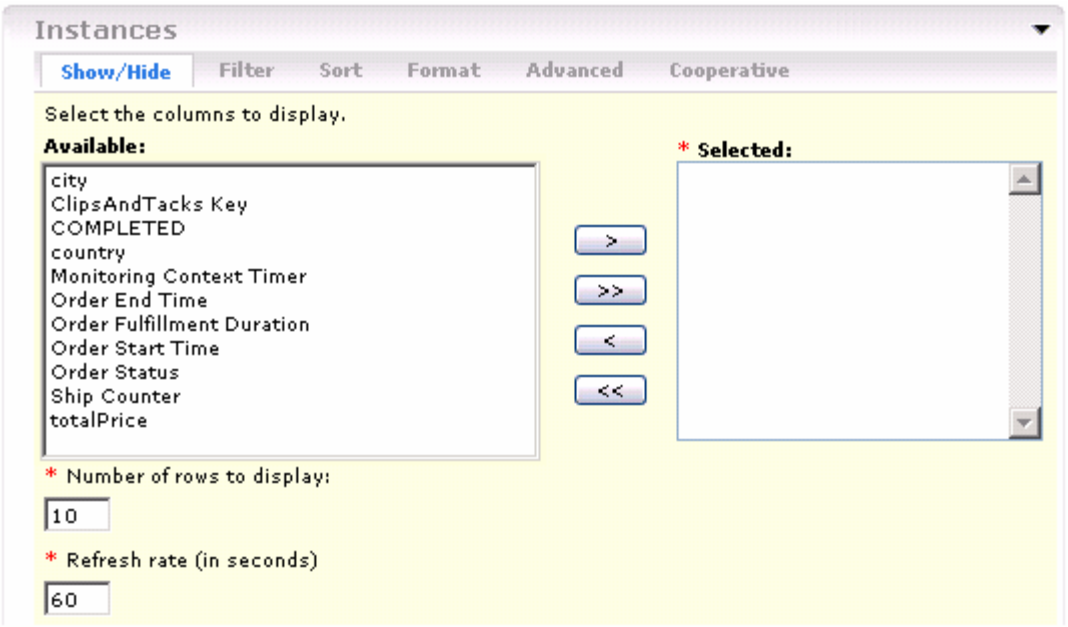


__ e. Click **New**, then enter a name for the dashboard for example **MyDashbaord**, then click **OK**.

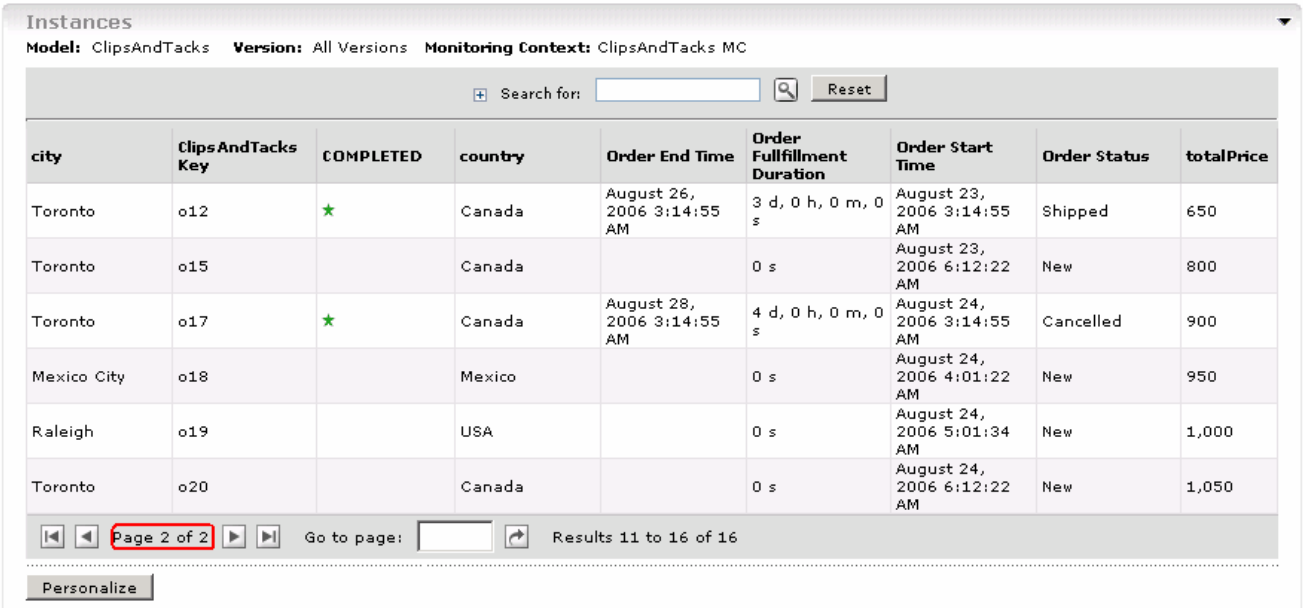


__ f. Click **Add to Dashboard**, then select **Instances**, then click **OK**. Note that you can also add a view by dragging the view from the palette on the right and dropping it onto the dashboard.

__ g. Click **Personalize**



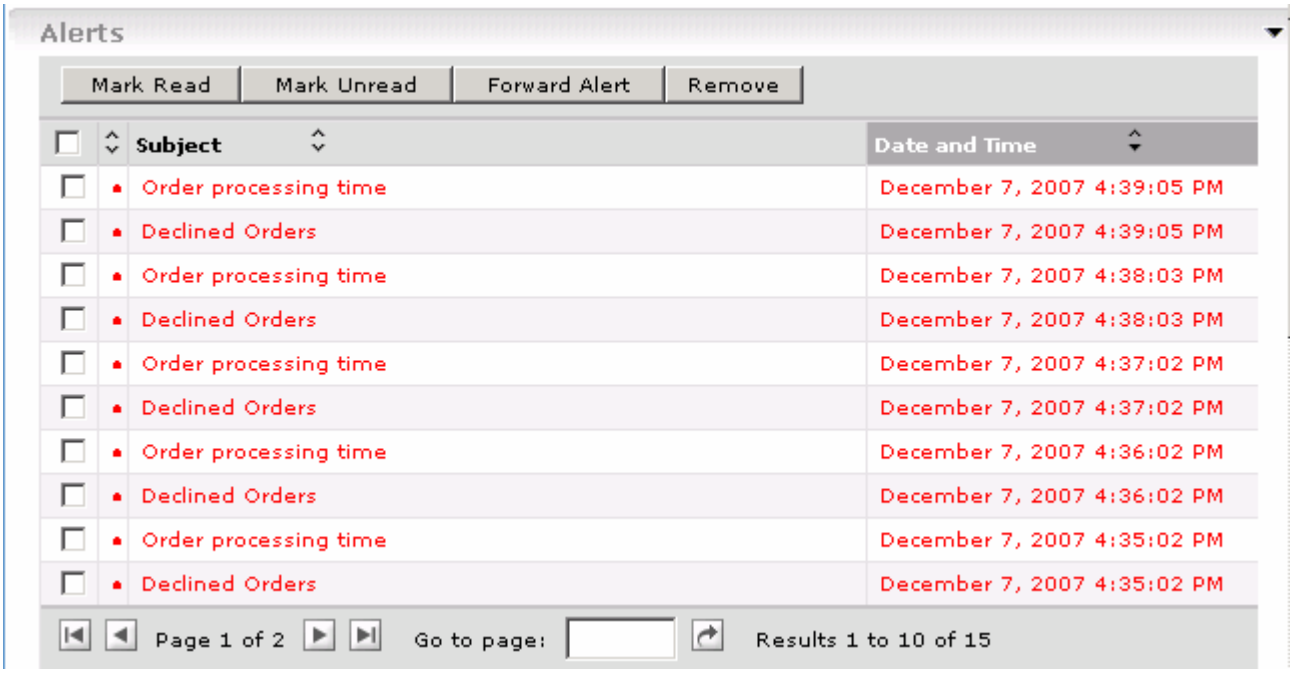
__ h. Click **>>** to copy all the metrics from the available list to the selected list. Then click **Save**. You should see a list of monitoring context instances for the events that you just processed.



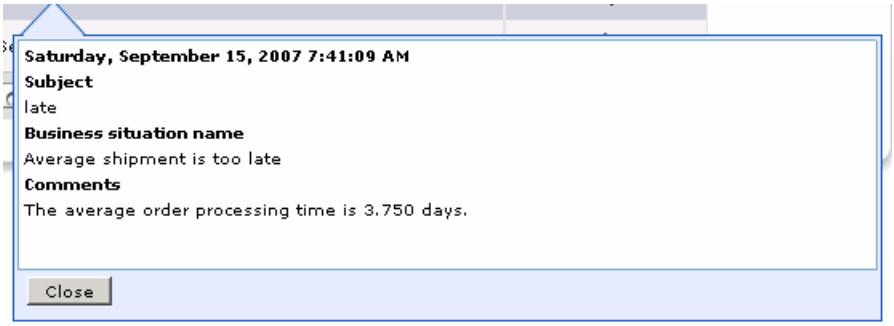
city	ClipsAndTacks Key	COMPLETED	country	Order End Time	Order Fullfillment Duration	Order Start Time	Order Status	totalPrice
Toronto	o12	★	Canada	August 26, 2006 3:14:55 AM	3 d, 0 h, 0 m, 0 s	August 23, 2006 3:14:55 AM	Shipped	650
Toronto	o15		Canada		0 s	August 23, 2006 6:12:22 AM	New	800
Toronto	o17	★	Canada	August 28, 2006 3:14:55 AM	4 d, 0 h, 0 m, 0 s	August 24, 2006 3:14:55 AM	Cancelled	900
Mexico City	o18		Mexico		0 s	August 24, 2006 4:01:22 AM	New	950
Raleigh	o19		USA		0 s	August 24, 2006 5:01:34 AM	New	1,000
Toronto	o20		Canada		0 s	August 24, 2006 6:12:22 AM	New	1,050

__ i. There are two pages of results, so if you browse through these instances, you see that there are 4 cancelled orders. Since you setup an alert if declined orders is greater than or equal to 3, then an alert should be displayed in the alert view.

__ j. Click **Add to Dashboard**, then select **Alerts**, then click **OK**. The alerts view is added to your dashboard, and you should see alerts listed since the model processed 3 or more cancelled orders, and the average order processing time is greater than 3 days.



__ k. You can click on an alert in the list and then you will get a pop-up showing the details of the alert.



What you did in this exercise

In the lab, you created a monitor model using Rational Application Developer or WebSphere Integration Developer.

Then you published the model to the monitor server.

You also configured action manager to create alerts based on business situation events.

You used the supplied program to simulate the submission of events from the monitored application.

You configured dashboards and viewed monitored data in several different types of views.

Appendix 1 – Import the solution into the monitor model editor

A solution has been provided so that you do not have to build the model from scratch. This section shows you how to import the monitor model into Rational Application Developer or WebSphere Integration Developer. After importing the model then you can proceed to the section to publish the model to the server.

____ 1. Start Rational Application Developer or WebSphere Integration Developer and setup the environment.

__ a. Start Rational Application Developer or WebSphere Integration Developer, and when prompted point to a new workspace such as **C:\workspaces\ClipsAndTacksXSD**

__ b. Close the Welcome tab

__ c. By default, you are in the Business Integration perspective. You want to open the Business Monitoring perspective. Select **Window → Open Perspective → Other → Show all → Business Monitoring**.

__ d. If it asks you to Confirm Enablement, then click **OK**.

____ 2. Create a new monitoring project.

__ a. Right click the Project Explorer and then select **New → Business Monitoring Project…**

__ b. For Project name, enter **ClipsAndTacks**

__ c. Click Finish. You will see the new project in the Project Explorer view.

____ 3. Import the supplied mm and cbe files.

__ a. Right click in the Project Explorer view, and then select **Import…**

__ b. Select **General → File system**, click **Next**.

__ c. Browse to the location containing the files, for example, **C:\Labfiles61\ClipsAndTacksXSD**

__ d. Select **ActivityEvent.xsd**

__ e. Select **ClipsAndTacks.mm**

__ f. Select **DeclinedOrderEvent.cbe**

__ g. Select **LateAverageOrderShippedEvent.cbe**

__ h. Select **NewOrderEvent.xsd**

__ i. Click **Finish**.

____ 4. Expand the project in the Project Explorer view, then expand the Event Definitions and you will see the new events listed. Expand Monitor Models and you will see the new ClipsAndTacks model listed.

____ 5. Now you may proceed to the section to publish the model to the server.