WebSphere Business Monitor V6.1 user defined functions lab

What this exercise is about

The objective of this lab is to show you how to use user defined functions in the monitor model. You will review the Java project that contains the user defined functions. You will edit the monitor model and add expressions that reference the user defined functions. You will also use the instances view to see the results in the dashboard.

Lab requirements

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

- Rational Application Developer 7.0 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 edit the monitor model to add references to user defined functions.
- Define a dashboard to view monitoring context instances.

Introduction

In this lab you will build on the Clips and Tacks model, which is described in another lab document titled 'WebSphere Business Monitor V6.1 Clips and Tacks business activity monitoring lab'. You will import the full Clips and Tacks model, then update the monitor model, deploy the changed model and submit events to the monitor model.

The user defined functions have been created for you so you can just import them into the workspace.

In the monitor model, you will define a new metric to track the cost of the ordering process. You will use an existing metric 'Order Fulfillment Duration,' which is the duration of the order process. You will use a user defined function to calculate the cost of the process using the order fulfillment duration as an input parameter. The cost function could retrieve the average unit cost from a database or another system, but for simplicity in this lab, the Java function is implemented using a random number generator.

URL Cheat sheet

These 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: Import the Clips and Tacks monitor model

In this section, you will import the Clips and Tacks monitor model into the workspace. If you already have the ClipsAndTacks model in your workspace after completing the Clips and Tacks lab, then you can skip this 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\ClipsAndTacks

__ b. Close the Welcome tab.

__ c. You want to open the Business Monitoring perspective. Select **Window > Open Perspective > 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, 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 on **ClipsAndTacks** 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\ClipsAndTacks

__ d. Select **ActivityEvent.cbe**

__ e. Select **ClipsAndTacks.mm**

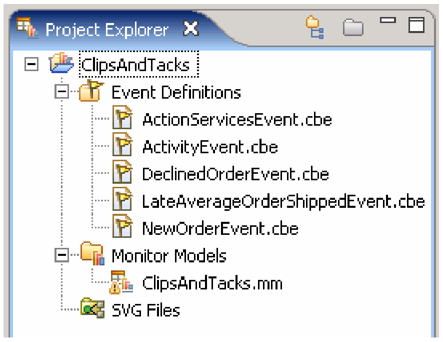
__ f. Select **DeclinedOrderEvent.cbe**

__ g. Select **LateAverageOrderShippedEvent.cbe**

__ h. Select **NewOrderEvent.cbe**

__ 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.


Project Explorer
ClipsAndTacks
Event Definitions
ActionServicesEvent.cbe
ActivityEvent.cbe
DeclinedOrderEvent.cbe
LateAverageOrderShippedEvent.cbe
NewOrderEvent.cbe
Monitor Models
ClipsAndTacks.mm
SVG Files

Part 2: Import the user defined function project

In this section you will use Rational Application Developer or WebSphere Integration Developer to import the project interchange containing the cost functions.

____ 1. Switch to the J2EE perspective.

__ a. Click **Window > Open Perspective > Other…**

__ b. Click **Show all**, then select **J2EE**

__ c. Click **OK**

____ 2. Import the cost function project interchange file.

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

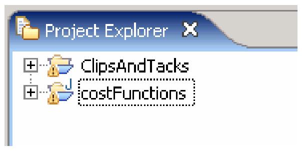
__ b. Select **Other** > **Project Interchange**, click **Next**.

__ c. For the .zip file, browse to the location containing the file, for example, c:\Labfiles61\UDF\CostFunctionPI.zip

__ d. Click **Select All**

__ e. Click **Finish**.

__ f. You will see the costFunctions project in the Project Explorer.

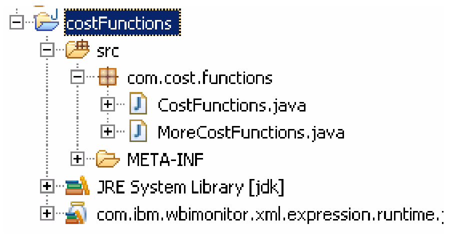


____ 3. Check for any errors in the Problems view. You should resolve any errors before continuing. Warnings and informational messages may be present but these are not a problem.

Part 3: Review the user defined functions and create the jar file

In this section you will review the user defined functions that you imported in the previous section. Then you will export a jar file that will be imported into the monitor model.

____ 1. Expand the costFunctions project in the Project Explorer.



____ 2. Double click on CostFunctions.java to open the Java editor.

____ 3. Notice the annotations in the file to specify the user defined functions to Monitor.

```
@XPathFunction(
        namespaceName = NAMESPACE,
        localName = "cost1",
        description = "calculate the cost based on task duration days",
```

____ 4. You will use the cost1 function in the monitor model.

____ 5. Take a look at the code in the cost1 function. It receives a parameter that is the number of day's duration for order fulfillment. It multiplies the duration by a random number between 0 and 1, and then multiplies that by 10. It returns this number as a decimal number, which is the calculated cost for the supplied duration. This is a simple example for the lab, but you could easily reference a database or Web service to retrieve the cost. Notice the Java types that are used for the function, argument and return types. Since you are implementing a Java method that will be used as an XPath function, you need to be aware of the mappings from Java to XML data types. Refer to the information center for all the available data types and mappings.

```
BigDecimal cost1(BigInteger numberDays) {
    // here you could get the cost from a database,
    // but we use a random function for simplicity for this example.
    // random returns a double between 0 and 1.
    return new BigDecimal(numberDays.doubleValue() * (random() * 10));
}
```

____ 6. There is another function cost2, which is defined in this file but is not used in the monitor model. There is also another file MoreCostFunctions.java, which contains other cost functions. This file is imported into the model to show how multiple files can be referenced in the workspace.

____ 7. Export the user defined functions to a jar file.

__ a. Right click on **costFunctions** folder (not the CostFunctions.java) in the Project Explorer, and select **Export…**

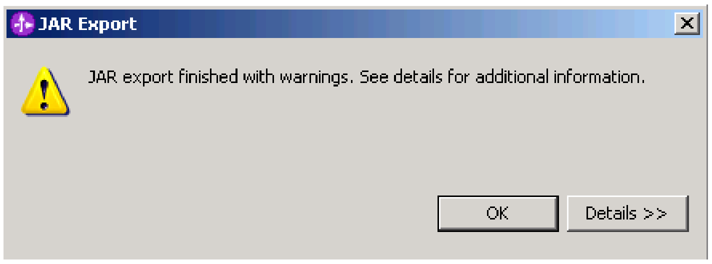
__ b. Select **Java > Jar** file, and click **Next**.

__ c. For the JAR file, click **Browse**… and navigate to a destination such as c:\temp.

__ d. Then specify a name for the jar file, such as costFunctions.jar, and click **Save**.

__ e. Click **Finish**.

__ f. You may get a popup indicating warnings, but this is fine, so just click **OK**.



JAR export finished with warnings. See details for additional information.

____ 8. 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 4: Import the user defined functions into the monitor model

In this section you will import the user defined functions jar into the monitor model.

____ 1. Switch to the Business Monitoring perspective.

__ a. Click **Window > Open Perspective > Business Monitoring**

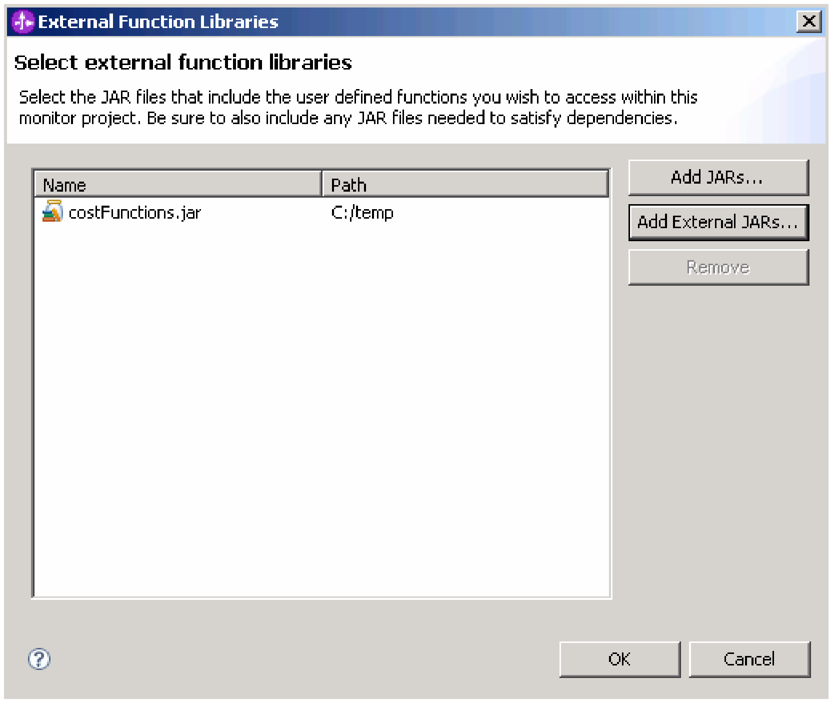
__ b. Click **OK**

____ 2. Define an external function library.

__ a. Right click on the **ClipsAndTacks** project in the project explorer, and then select **External Function Libraries…**

__ b. Click **Add External JARs…**

__ c. Navigate to the location of your function jar, for example, c:\temp\costFunctions.jar, then click Open


External Function Libraries

Select external function libraries

Select the JAR files that include the user defined functions you wish to access within this monitor project. Be sure to also include any JAR files needed to satisfy dependencies.

Name	Path
costFunctions.jar	C:/temp

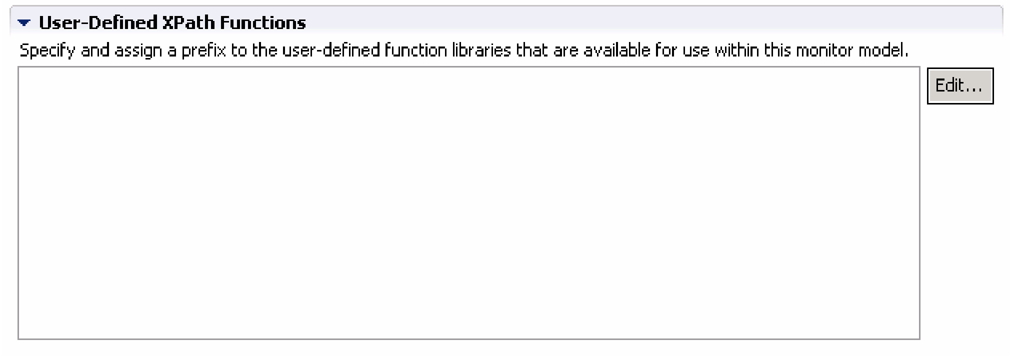
__ d. Click **OK**.

____ 3. Define the cost functions to the monitor model.

__ a. Double click on **ClipsAndTacks > Monitor Models > ClipsAndTacks.mm** in the project explorer.

__ b. The monitor model editor opens.

__ c. On the Monitor Model Details tab, notice the area for user defined functions:

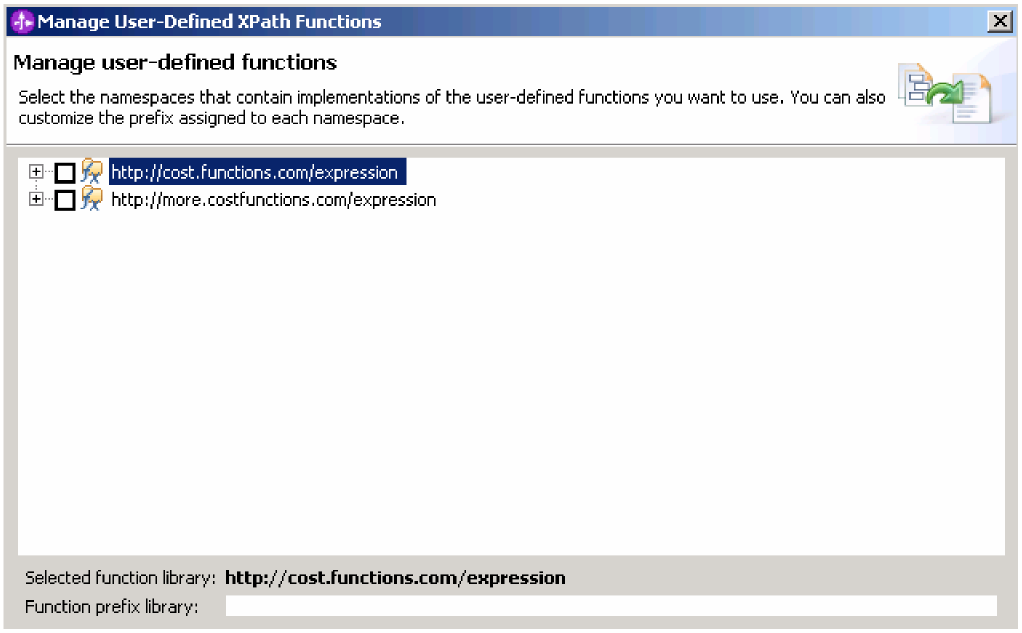


User-Defined XPath Functions

Specify and assign a prefix to the user-defined function libraries that are available for use within this monitor model.

__ d. Click **Edit**…

__ e. You should see a list of the function files in the library.



Manage User-Defined XPath Functions

Manage user-defined functions

Select the namespaces that contain implementations of the user-defined functions you want to use. You can also customize the prefix assigned to each namespace.

http://cost.functions.com/expression

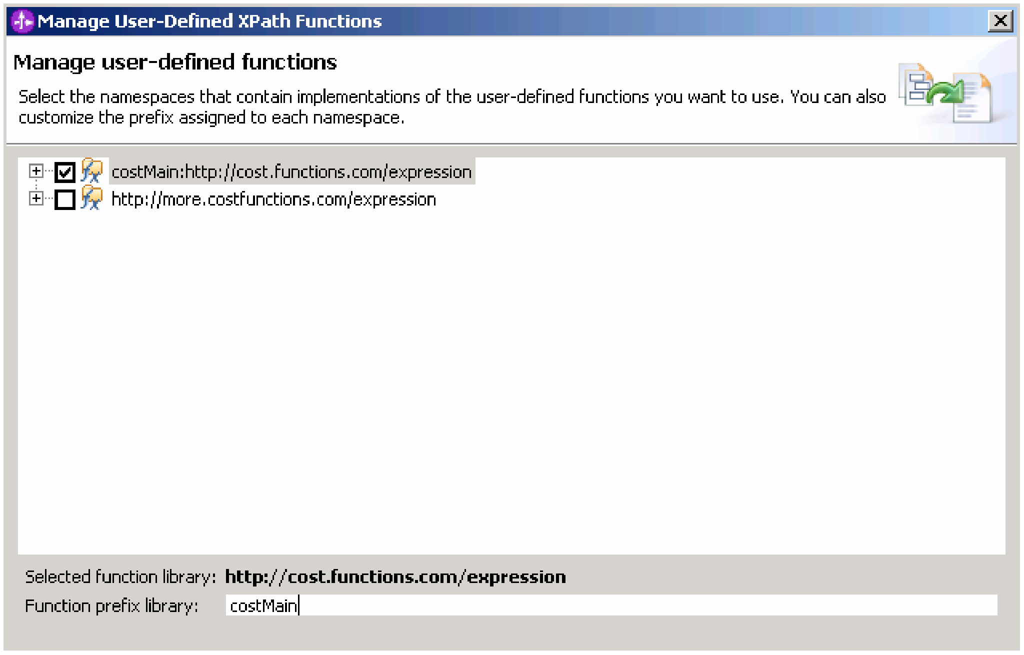
http://more.costfunctions.com/expression

Selected function library: **http://cost.functions.com/expression**

Function prefix library:

__ f. You need to assign a prefix to each namespace.

__ g. Check the box for the first one in the list, then type a name for the function prefix down below, for example, **costMain**.



Manage User-Defined XPath Functions

Manage user-defined functions

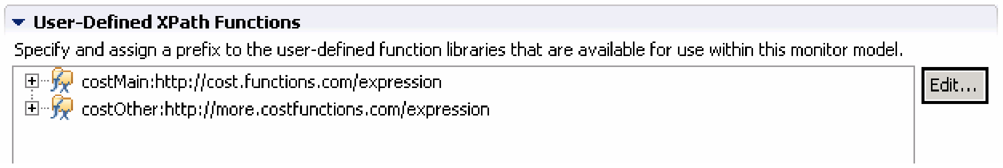
Select the namespaces that contain implementations of the user-defined functions you want to use. You can also customize the prefix assigned to each namespace.

[✓] costMain:http://cost.functions.com/expression
[ ] http://more.costfunctions.com/expression

Selected function library: **http://cost.functions.com/expression**
Function prefix library: costMain

__ h. Check the box for the second one in the list, then type a name for the function prefix down below, for example, **costOther**.

__ i. Click **Finish**, then you should see the two listings on the Monitor Model Details tab in the user defined functions list.



User-Defined XPath Functions

Specify and assign a prefix to the user-defined function libraries that are available for use within this monitor model.

costMain:http://cost.functions.com/expression
costOther:http://more.costfunctions.com/expression

__ j. Now the functions are available for use in the monitor model.

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

__ l. 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 5: Add expressions that reference user defined functions

In this section you will use Rational Application Developer or WebSphere Integration Developer to edit the monitor model. You will create a new metric and add expressions that use the cost functions. You will be calculating the cost based on the order fulfillment duration metric, which is already defined in the model. Since the order fulfillment duration metric is calculated based on two triggers, one for the order shipment and one for the order cancellation, you will use the same triggers to fire the calculation of this cost metric.

____ 1. If not already open, switch to the Business Monitoring perspective.

____ 2. Double click on the monitor model **ClipsAndTacks.mm** in the project explorer to open the model editor.

____ 3. Create a new metric in the monitor model for order fulfillment cost.

__ a. In the model navigator in the model editor, right click on 'ClipsAndTacks MC', then select New > Metric.

__ b. For name, enter: Order Fulfillment Cost

__ c. For ID, accept default: Order_Fulfillment_Cost

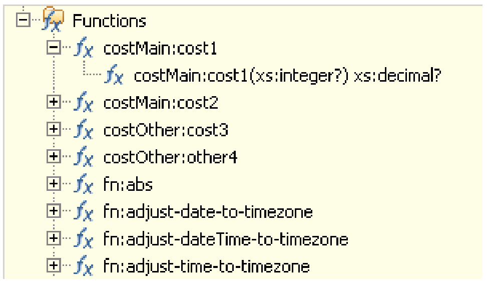
__ d. For type, select: Decimal

__ e. Click OK

__ f. For Metric Value Expressions, click Add

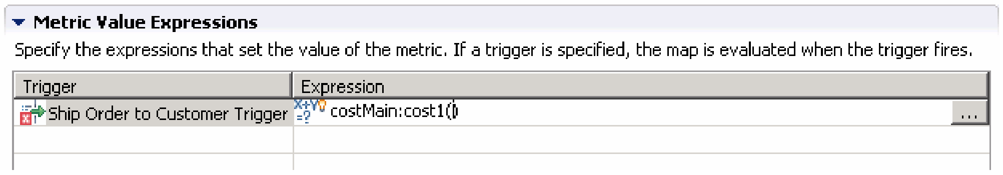
__ g. In the first row of the Metric Value Expressions table, select the cell under column Trigger, then a button is shown, then click the button and select Trigger type > ClipsAndTacks MC > Ship Order to Customer Trigger, click OK

__ h. In the first row of the Metric Value Expressions table, select the cell under column Expression, then bring up content assist by pressing Ctrl-space. Expand Functions and notice that your cost user defined functions are listed here with the prefixes costMain and costOther that you defined earlier:



```
Functions
  costMain:cost1
    costMain:cost1(xs:integer?) xs:decimal?
  costMain:cost2
  costOther:cost3
  costOther:other4
  fn:abs
  fn:adjust-date-to-timezone
  fn:adjust-dateTime-to-timezone
  fn:adjust-time-to-timezone
```

__ i. Select costMain:cost1 and it is added to the expression.



Metric Value Expressions

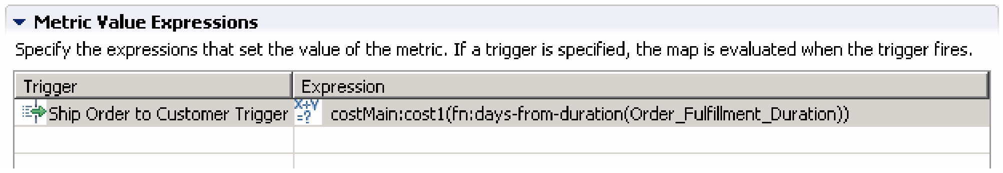
Specify the expressions that set the value of the metric. If a trigger is specified, the map is evaluated when the trigger fires.

Trigger	Expression
Ship Order to Customer Trigger	costMain:cost1()

__ j. Now you need to define the integer duration input to the function. There is a metric in the model, Order Fulfillment Duration, that you can use, but it is type duration so you will also need to use a function to convert the duration to an integer number of days. Use content assist, or type the expression manually so that it looks like this:

costMain:cost1(fn:days-from-duration(Order_Fulfillment_Duration))

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



Metric Value Expressions

Specify the expressions that set the value of the metric. If a trigger is specified, the map is evaluated when the trigger fires.

Trigger	Expression
Ship Order to Customer Trigger	costMain:cost1(fn:days-from-duration(Order_Fulfillment_Duration))

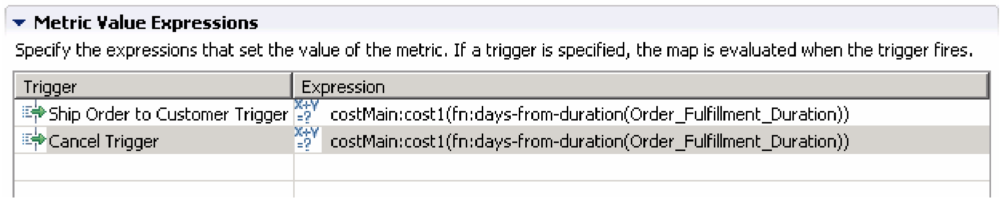
__ l. For Metric Value Expressions, click Add

__ m. 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

__ n. In the second row of the Metric Value Expressions table, select the cell under column Expression. Use content assist, or type the same expression as above so that it looks like this:

costMain:cost1(fn:days-from-duration(Order_Fulfillment_Duration))

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



Metric Value Expressions

Specify the expressions that set the value of the metric. If a trigger is specified, the map is evaluated when the trigger fires.

Trigger	Expression
Ship Order to Customer Trigger	costMain:cost1(fn:days-from-duration(Order_Fulfillment_Duration))
Cancel Trigger	costMain:cost1(fn:days-from-duration(Order_Fulfillment_Duration))

____ 4. 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 6: Publish the model to the server

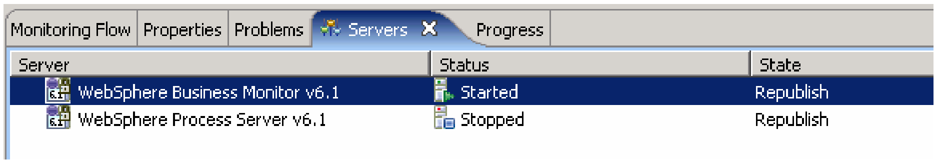
In this section you will use Rational Application Developer or WebSphere Integration Developer to publish the monitor model to the monitor server.

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

____ 2. If you have already generated the ClipsAndTacks model for another lab, then you may need to select 'overwrite existing projects'. Then click **Finish**.

____ 3. 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 might be warnings, but there should not be any errors. If you see errors, then try to perform a clean to see if the errors can be removed: Project > Clean… > select 'Clean all projects', then click OK.

____ 4. Click the Servers tab, then right click and select the **Start** option to start the server WebSphere Business Monitor V6.1. This may take a few minutes to complete.



Server	Status	State
WebSphere Business Monitor v6.1	Started	Republish
WebSphere Process Server v6.1	Stopped	Republish

____ 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. Click Finish. Note: If the application is already in the list of configured projects, then you may have run another lab and published this application already, so in that case, cancel this window and select the 'Publish' option from the server menu.

____ 7. A progress message is displayed in the lower right corner of the window.

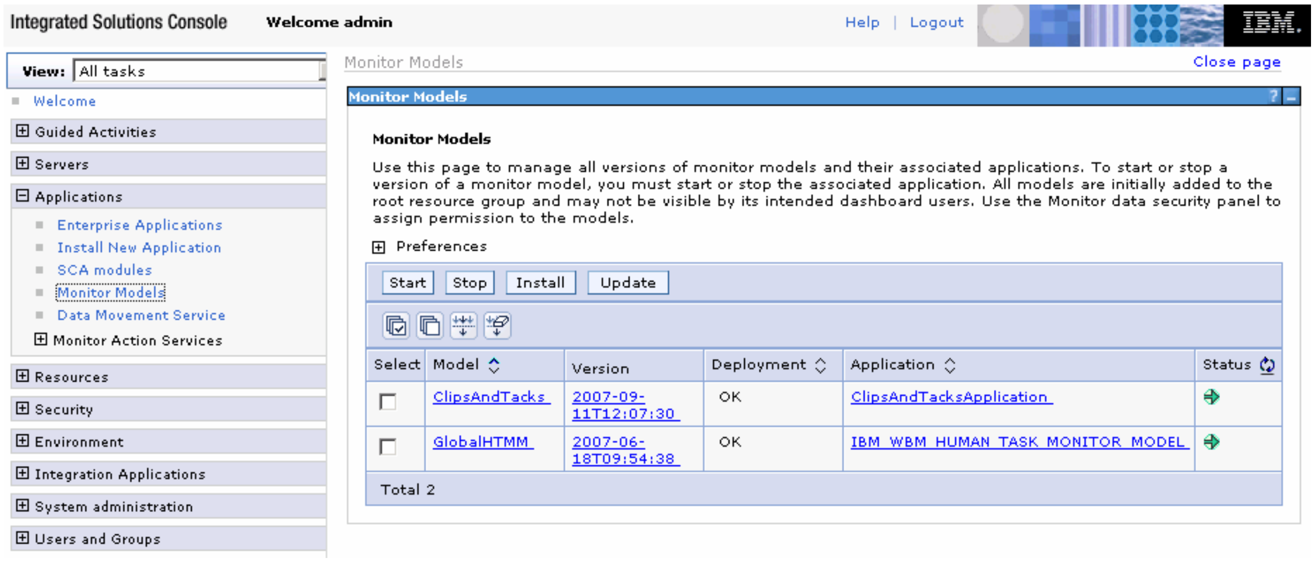
____ 8. Check the messages in the console view. You should see this message when the application has been started:

Application started: ClipsAndTacksApplication

____ 9. In the servers view, right click, then select **Run administrative console**. You should see it open in a separate tab. It will prompt you for user ID and password. Enter 'admin' without the quotation marks for user ID, and then enter 'admin' without the quotes for password.

____ 10. Click **Log in**

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



Monitor Models

Use this page to manage all versions of monitor models and their associated applications. To start or stop a version of a monitor model, you must start or stop the associated application. All models are initially added to the root resource group and may not be visible by its intended dashboard users. Use the Monitor data security panel to assign permission to the models.

Select	Model	Version	Deployment	Application	Status
☐	ClipsAndTacks	2007-09-11T12:07:30	OK	ClipsAndTacksApplication	➔
☐	GlobalHTMM	2007-06-18T09:54:38	OK	IBM_WBM_HUMAN_TASK_MONITOR_MODEL	➔
Total 2					

____ 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.

____ 14. If you are using the integrated server within WebSphere Integration Developer or Rational Application Developer, then you do not need to setup Monitor data security, since the admin user is automatically authorized to all models. If you are using a different server, then you should open the administrative console, navigate to **Security > Monitor Data Security,** then add the model, role and user information to a resource group.

Part 7: Run events to exercise the model

Rather than installing a Java EE 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 that 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\AppServer'. If you are using the integrated monitor server in WebSphere Integration Developer, the path is <WID-Install-Path>\pf\WBMonSrv_wps. If you are using the integrated monitor server in Rational Application Developer, the path is <RAD-Install-Path>\pf\WBMonSrv.

____ 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\AppServer\profiles\WBMon01\logs\server1\SystemOut.log . For the integrated server in WebSphere Integration Developer, the path is <WID-Install-Path>\pf\WBMonSrv_wps\logs\server1\SystemOut.log and the portNumber is 2810. For the integrated server in Rational Application Developer, the path is <RAD-Install-Path>\pf\WBMonSrv\logs\server1\SystemOut.log and the portNumber is 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

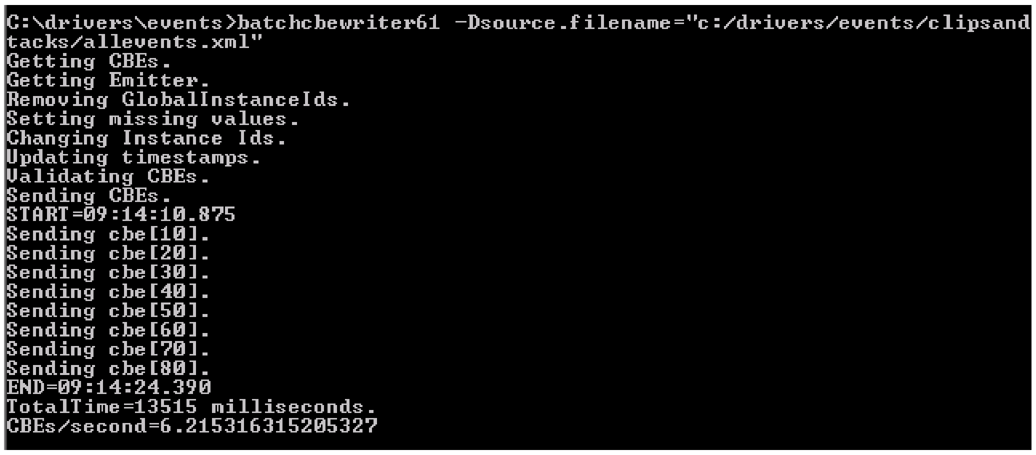
__ a. cd \Labfiles61\ClipsAndTacks\BatchCBEWriter

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

__ a. batchcbewriter61 -Dsource.filename="c:/labfiles61/clipsandtacks/cbe/allevents.xml"

__ b. When it prompts you for user identity and password, enter 'admin' for both (without the quotation marks)

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


```
C:\drivers\events>batchcbewriter61 -Dsource.filename="c:/drivers/events/clipsand
tacks/allevents.xml"
Getting CBEs.
Getting Emitter.
Removing GlobalInstanceIds.
Setting missing values.
Changing Instance Ids.
Updating timestamps.
Validating CBEs.
Sending CBEs.
START=09:14:10.875
Sending cbe[10].
Sending cbe[20].
Sending cbe[30].
Sending cbe[40].
Sending cbe[50].
Sending cbe[60].
Sending cbe[70].
Sending cbe[80].
END=09:14:24.390
TotalTime=13515 milliseconds.
CBEs/second=6.215316315205327
```

Part 8: Create a dashboard

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

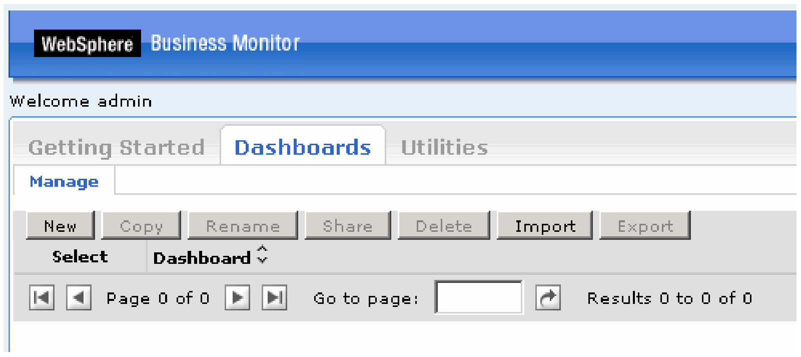
____ 1. Open the dashboard.

__ a. In Rational Application Developer or WebSphere Integration Developer, click Window > Web Browser. The default browser is 'Internal Web Browser', but you should not use this one since some standard functions are not provided that you may need. Select 'Default system Web browser' or any other listed browser other than the internal browser.

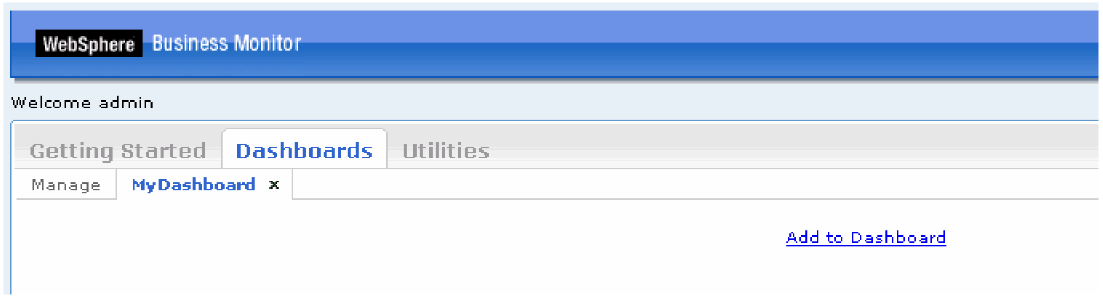
__ b. In WebSphere Integration Developer, in the servers view, right click and select **WebSphere Business Monitor Dashboard**

__ c. When prompted, enter **admin** for the user ID and enter **admin** for the password. You must log in with 'admin' so that you can view the alerts that were setup in action manager to be sent to this particular user ID. Also, in the toolkit environment, this is the user that is automatically defined on the secured server.

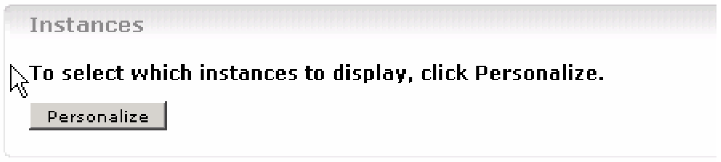
__ d. Click the **Dashboards** tab.



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



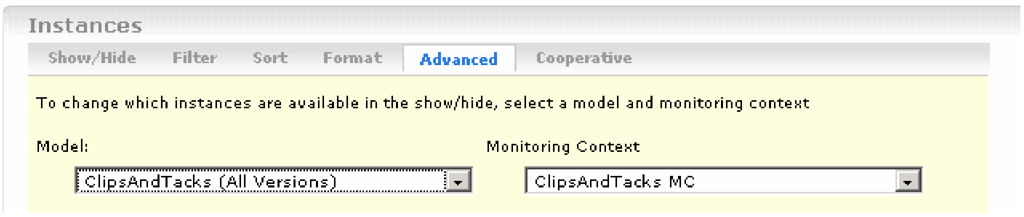
__ f. Click **Add to Dashboard**, then select **Instances**, then click **OK**. Note that you can also add a view by dragging the view from the palette on the right and dropping it onto the dashboard.


Instances

To select which instances to display, click Personalize.

__ g. Click **Personalize**.

__ h. Click the **Advanced** tab, and select model 'ClipsAndTacks (All Versions)'



Instances

Show/Hide | Filter | Sort | Format | Advanced | Cooperative

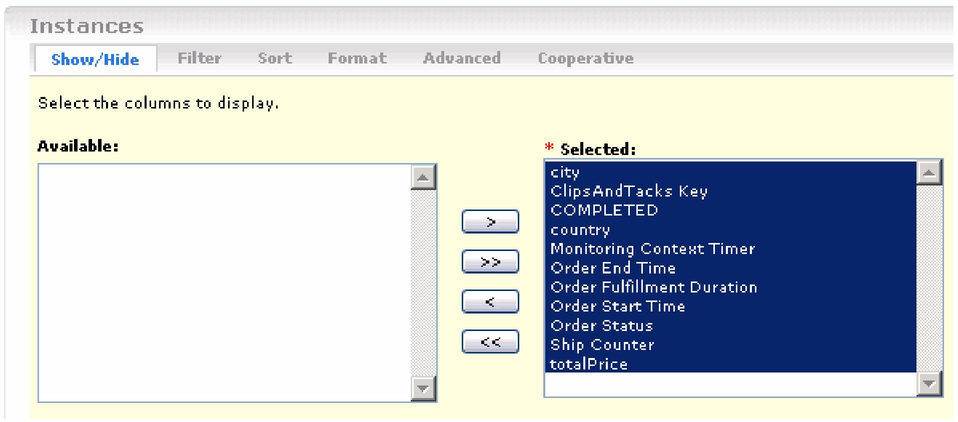
To change which instances are available in the show/hide, select a model and monitoring context

Model: ClipsAndTacks (All Versions)

Monitoring Context: ClipsAndTacks MC

__ i. Click the **Show/Hide** tab.

__ j. Click **>>** to copy all the metrics from the available list to the selected list.



Instances

Show/Hide | Filter | Sort | Format | Advanced | Cooperative

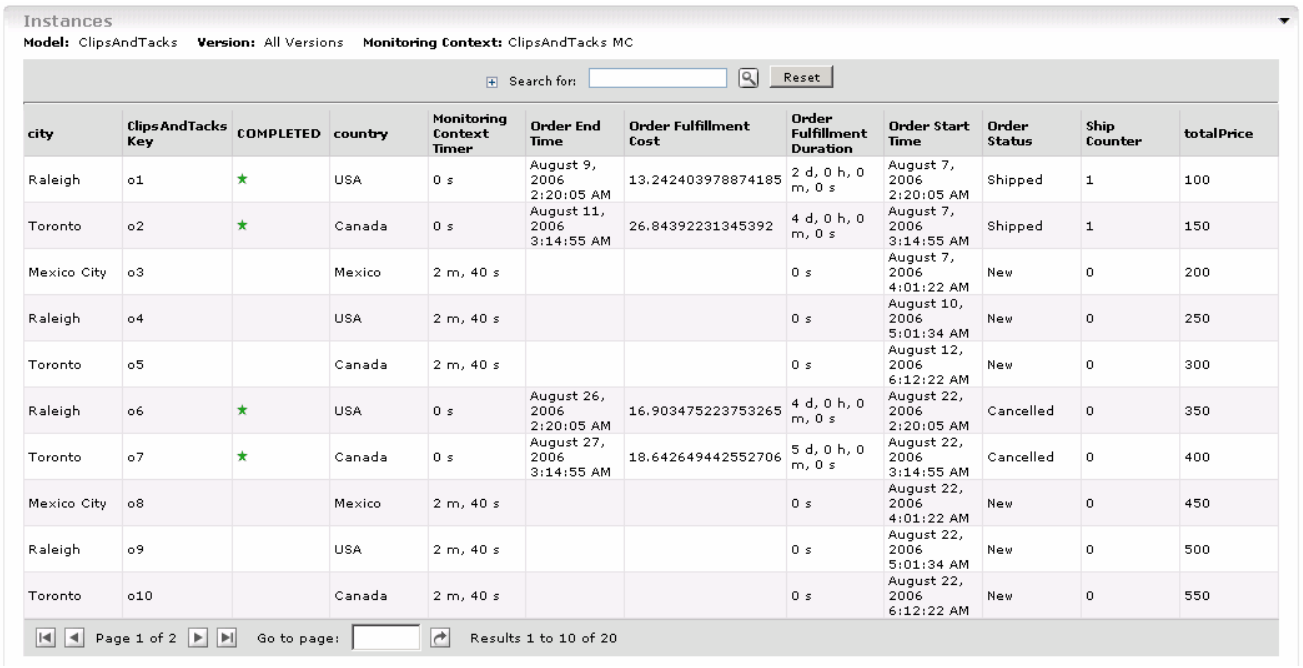
Select the columns to display.

Available:

* Selected:

city
ClipsAndTacks Key
COMPLETED
country
Monitoring Context Timer
Order End Time
Order Fulfillment Duration
Order Start Time
Order Status
Ship Counter
totalPrice

__ k. Then click **OK**. You should see a list of monitoring context instances for the events that you just processed.



Instances

Model: ClipsAndTacks **Version:** All Versions **Monitoring Context:** ClipsAndTacks MC

city	ClipsAndTacks Key	COMPLETED	country	Monitoring Context Timer	Order End Time	Order Fulfillment Cost	Order Fulfillment Duration	Order Start Time	Order Status	Ship Counter	totalPrice
Raleigh	o1	★	USA	0 s	August 9, 2006 2:20:05 AM	13.242403978874185	2 d, 0 h, 0 m, 0 s	August 7, 2006 2:20:05 AM	Shipped	1	100
Toronto	o2	★	Canada	0 s	August 11, 2006 3:14:55 AM	26.84392231345392	4 d, 0 h, 0 m, 0 s	August 7, 2006 3:14:55 AM	Shipped	1	150
Mexico City	o3		Mexico	2 m, 40 s			0 s	August 7, 2006 4:01:22 AM	New	0	200
Raleigh	o4		USA	2 m, 40 s			0 s	August 10, 2006 5:01:34 AM	New	0	250
Toronto	o5		Canada	2 m, 40 s			0 s	August 12, 2006 6:12:22 AM	New	0	300
Raleigh	o6	★	USA	0 s	August 26, 2006 2:20:05 AM	16.903475223753265	4 d, 0 h, 0 m, 0 s	August 22, 2006 2:20:05 AM	Cancelled	0	350
Toronto	o7	★	Canada	0 s	August 27, 2006 3:14:55 AM	18.642649442552706	5 d, 0 h, 0 m, 0 s	August 22, 2006 3:14:55 AM	Cancelled	0	400
Mexico City	o8		Mexico	2 m, 40 s			0 s	August 22, 2006 4:01:22 AM	New	0	450
Raleigh	o9		USA	2 m, 40 s			0 s	August 22, 2006 5:01:34 AM	New	0	500
Toronto	o10		Canada	2 m, 40 s			0 s	August 22, 2006 6:12:22 AM	New	0	550

Page 1 of 2 Go to page: Results 1 to 10 of 20

____ 2. Notice the new metric Order Fulfillment Cost. For shipped or cancelled orders, you see that the cost1 function has been used to calculate the cost based on the Order Fulfillment Duration.

What you did in this exercise

In the lab, you imported user defined functions into a monitor model, and you created a cost metric that referenced the functions in expressions.

Then you published the model to the monitor server.

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

You configured dashboards to view the new cost metric in the instances view.

This page is left intentionally blank.