Est. time: 3 hours (less if experienced with WebSphere Message Broker or WebSphere ESB)

IBM WebSphere DataPower XC10: **Elastic caching solutions for your ESB**

What this exercise is about

The objective of this lab is to provide examples of using the IBM WebSphere DataPower XC10 caching appliance within the Enterprise Service Bus framework, specifically in WebSphere Message Broker and WebSphere Enterprise Service Bus.

This lab is provided AS-IS, with no formal IBM support.

Introduction

IT integration is the implementation of Service-Oriented Architectures (SOAs) using web services technologies, and the concept of an Enterprise Service Bus (ESB) has been expressed as a key component of the SOA infrastructure. An ESB is a set of infrastructure capabilities implemented by middleware technology that enable an SOA. A capability model for an ESB typically includes communications, service interaction, integration, quality of services, security, service level, message processing, management and autonomic, modeling, and infrastructure intelligence.

Customers will have different capability requirements, and thus various solutions have been developed by IBM to provide for these different levels of capabilities. Two products that are foundational as ESBs are WebSphere Message Broker and WebSphere Enterprise Service Bus.

The IBM WebSphere DataPower XC10 Appliance is an easy-to-use caching appliance. It provides simplified deployment at the caching tier of your enterprise application infrastructure. Client code is provided that easily integrates non-intrusively into existing applications, whether running on a WebSphere Application Server or running stand-alone applications.

This lab will provide examples of how the XC10 client code can easily integrate non-intrusively into the two foundational ESB solutions mentioned above, into a WebSphere Message Broker JavaCompute node and into a WebSphere ESB mediation flow.

This lab assumes some basic working knowledge of the XC10 web console, and that the simple data grid is already created. It does not provide any product installation instructions; it assumes you will have the necessary products appropriately installed in a Windows environment.

What you should be able to do

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

- Create and run a stand-alone Java application that exercises the basic essentials of accessing a data grid on the XC10 appliance.
- Drop the same Java code into an existing WebSphere Message Broker JavaCompute node sample and access a data grid on the XC10 appliance from the message flow.
- Use ObjectGrid primitives to modify an existing WebSphere Enterprise Service Bus sample to access a data grid on the XC10 appliance.

Lab requirements

There are three parts to this lab, and each part integrates the XC10 appliance with different products. Thus the WebSphere DataPower XC10 appliance and the eXtreme Scale client code are required for all three parts, while other product requirements are unique to each part.

All three parts require:

- (hardware) IBM WebSphere DataPower XC10 appliance with latest firmware level (while it is preferred to use a V2 appliance, a V1 appliance will suffice).
- (software) WebSphere eXtreme Scale Client V7.1 with latest client fix pack.

Additional unique software is required for each Part as follows.

Part 1:

- Eclipse IDE for Java EE Developers (Helios)

Part 2:

- WebSphere MQ V7.1.0.6
- WebSphere Message Broker V7.0.0.3
- WebSphere Message Broker Toolkit V7.0.0.3

Part 3:

- IBM Integration Designer Version 7.5.0.0
- IBM WebSphere Application Server - Network Deployment Version 7.0.0.17
- IBM WebSphere Application Server V7 Feature Pack for Service Component Architecture (SCA) Version 1.0.1.11
- IBM WebSphere Application Server V7 Feature Pack for XML Version 1.0.0.9

- IBM DB2 Express 32 bit Version 9.7.0.4
- IBM Business Process Manager Advanced V7.5 - Process Server Version 7.5.0.0

Part 1: Develop and run a stand-alone Java application accessing a data grid

Part 1 of this lab shows how to create and run a simple stand-alone Java application that accesses a data grid using the Java ObjectGrid API. The subsequent Part 2 section will then demonstrate dropping the same Java code into an existing WebSphere Message Broker JavaCompute node sample.

The Helios version of Eclipse IDE for Java EE Developers is used in this exercise for developing and running the stand-alone Java application, which was downloaded from:

http://www.eclipse.org/downloads/packages/release/helios/sr2.

The Java application demonstrated in this section is in Appendix A. This Java application called SimpleGridDemo performs the basic functions of:

- Connecting to the XC10 collective,
- Obtaining an ObjectGrid instance
- Creating a session
- Obtaining an ObjectMap
- Accessing data in the ObjectMap (get, remove, insert)
- Implicitly disconnecting.

Some observations about this demonstration Java application:

- Security credentials are provided in this demo, but if they are not wanted, the following (commented) connect statement in the demo illustrates how to connect without credentials:

```
//                              ccc = ogm.connect(hostName+":2809", null, null);
```

- This application will use the default ObjectMap name which is the same as the grid name. Optionally, if the use of eviction and locking strategies is required, you can use dynamic ObjectMap names as illustrated by the following commented lines:

```
//     static String mapName  = "Demo_simple.LAT.P";
//            map.setTimeToLive(60);
```

- If you want to group multiple operations within a transaction, then you can use the ***begin*** and ***commit*** methods as illustrated by the following commented lines:

```
//            sess.begin();
//            sess.commit();
```

____ *1.* Start Java EE Eclipse and ensure you are on the **Package Explorer** view.

____ 2. Select **File**, **New**, and **Java Project**.

____ 3. In the **Create a Java Project** window:

__ a. Give a **Project Name** of SimpleGridDemo**.**

__ b. Ensure you have selected your JRE of choice.

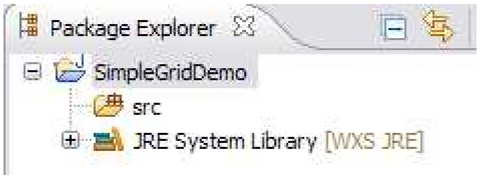
NOTE: The ORB provided by the SUN JRE implementation does not work with WebSphere eXtreme Scale at this time. Use an IBM JRE. For further information, see: http://publib.boulder.ibm.com/infocenter/wxsinfo/v7r1/index.jsp?topic=%2Fcom.ibm.websphere.extremescale.admin.doc%2Ftxscfgorb.html).

__ c. Select Finish.

____ 4. Add the WebSphere eXtreme Scale ObjectGrid Java library to this project.

NOTE: This step assumes you already have the WebSphere eXtreme Scale client installed locally in stand-alone mode (the WebSphere Application Server version of the client will not work in this configuration). If assistance is required installing the WebSphere eXtreme Scale client, see the "Client Installation" module here: http://publib.boulder.ibm.com/infocenter/ieduasst/v1r1m0/topic/com.ibm.iea.wdatapower/wdatapower/2.0/xc10.html?dmuid=20110602164106460770.

__ a. Expand the "+" on the SimpleGridDemo project:



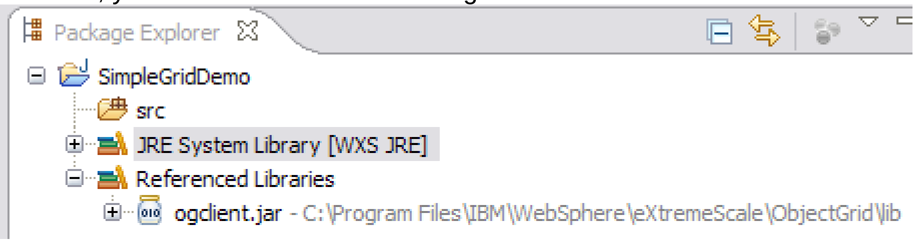
__ b. Right-click **JRE System Library**, select **Build Path**, and select **Configure Build Path**.

__ c. Select the **Libraries** tab, and select **Add External JARS**.

__ d. You will need to add the WebSphere eXtreme Scale ogclient.jar file here (the wsogclient.jar will NOT work for this configuration). If you do not have this jar file available, you will need to download the stand-alone client from Fix Central: http://www-933.ibm.com/support/fixcentral/swg/selectFixes?parent=ibm/WebSphere&product=ibm/WebSphere/WebSphere+eXtreme+Scale&release=All&platform=All&function=fixId&fixids=7.1.0.3-WS-WXS-Client-FP0000003&includeSupersedes=0

__ e. With a stand-alone WebSphere eXtreme Scale client installed, navigate to your >**root**<eXtremeScale\ObjectGrid\lib directory and select the **ogclient.jar** file.

__ f. Click **OK**; you should now see something similar to:



Package Explorer

SimpleGridDemo
src
JRE System Library [WXS JRE]
Referenced Libraries
ogclient.jar - C:\Program Files\IBM\WebSphere\eXtremeScale\ObjectGrid\lib

____ **5.** Create your source Java application:

__ a. Right-click src, select New, and select Class.

__ b. Enter SimpleGridDemo as the class name, and select Finish.

__ c. In the editor, replace the existing code snippet with the Java code in Appendix A.

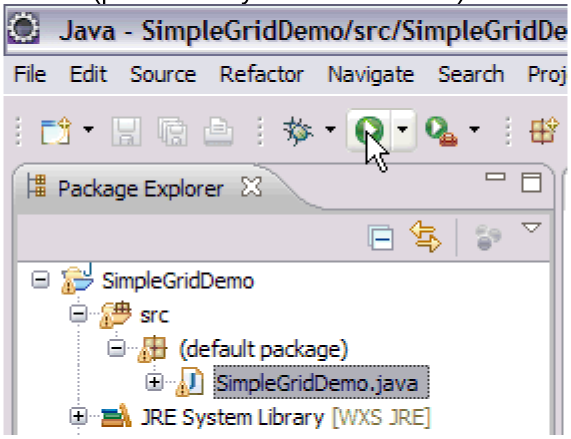
NOTE: This Java application imports more ObjectGrid packages then required, and are provided for illustration purposes. For further documentation on all the ObjectGrid client packages available, see: http://publib.boulder.ibm.com/infocenter/wdpxc/v2r0/topic/com.ibm.websphere.datapower.xc.javadoc.doc/topics/overview-summary.html.

____ 6. Test the stand-alone Java program:

__ a. Ensure you have a simple data grid created on the XC10 collective with the name given in the Java application.

NOTE: If assistance is required to create the data grid, see the “Data Grid Overview” module here: http://publib.boulder.ibm.com/infocenter/ieduasst/v1r1m0/topic/com.ibm.iea.wdatapower/wdatapower/2.0/xc10.html?dmuid=20110602164106460770.

__ b. Ensure the Java application is selected (as below) and click the green right arrow in the toolbar (pointed to by the cursor below):



Java - SimpleGridDemo/src/SimpleGridDe

Package Explorer

SimpleGridDemo
src
(default package)
SimpleGridDemo.java
JRE System Library [WXS JRE]

__ c. Via the web console of the XC10 appliance, monitor the data grid to observe the various result statistics for the data grid.

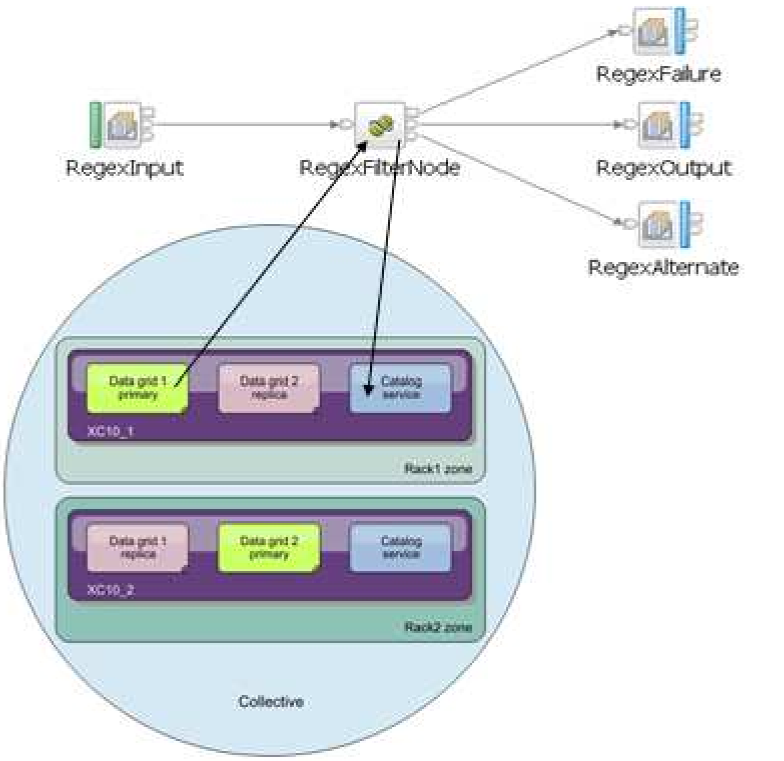
NOTE: If assistance is required for monitoring the data grid, see the “Appliance console – Monitoring” module here: http://publib.boulder.ibm.com/infocenter/ieduasst/v1r1m0/topic/com.ibm.iea.wdatapower/wdatapower/2.0/xc10.html?dmuid=20110602164106460770.

Part 2: Implement the ObjectGrid API into a WebSphere Message Broker JavaCompute node

In Part 1 of this lab, you saw how to create a simple Java program to access a data grid on an XC10 collective. In Part 2, you will now go through the process of implementing very similar Java code in an existing sample of a WebSphere Message Broker JavaCompute node, namely in the sample RegexFilterNode message flow.

Note that for purposes of this lab, the calls made to the XC10 data grid have no relationship to the existing code in the JavaCompute node; they demonstrate how the APIs are typically used.

WebSphere Message Broker RegexFilterNoteFlow msg flow with DataPower XC10:

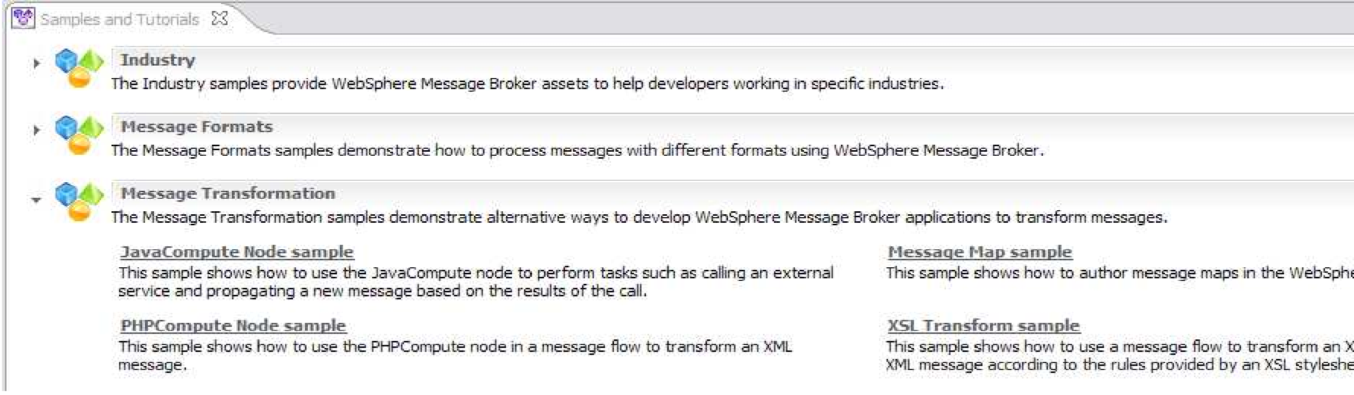


This exercise assumes you already have WebSphere MQ, WebSphere Message Broker, and the WebSphere Message Broker Toolkit installed (on Windows). This exercise was created using WebSphere MQ V7.0.1.6, WebSphere Message Broker V7.0.0.3, and Toolkit V7.0.0.3, but all supported versions should work the same.

____ *7.* Start the WebSphere Message Broker Toolkit.

____ *8.* You should see either "**Samples and Tutorials**" or "**Start from samples**" in the Toolkit. Scroll down the samples until you come to the "**Message Transformation**" set. Click the twistie to display all the

"**Message Transformation**" samples.



Industry
The Industry samples provide WebSphere Message Broker assets to help developers working in specific industries.

Message Formats
The Message Formats samples demonstrate how to process messages with different formats using WebSphere Message Broker.

Message Transformation
The Message Transformation samples demonstrate alternative ways to develop WebSphere Message Broker applications to transform messages.

JavaCompute Node sample
This sample shows how to use the JavaCompute node to perform tasks such as calling an external service and propagating a new message based on the results of the call.

Message Map sample
This sample shows how to author message maps in the WebSphe

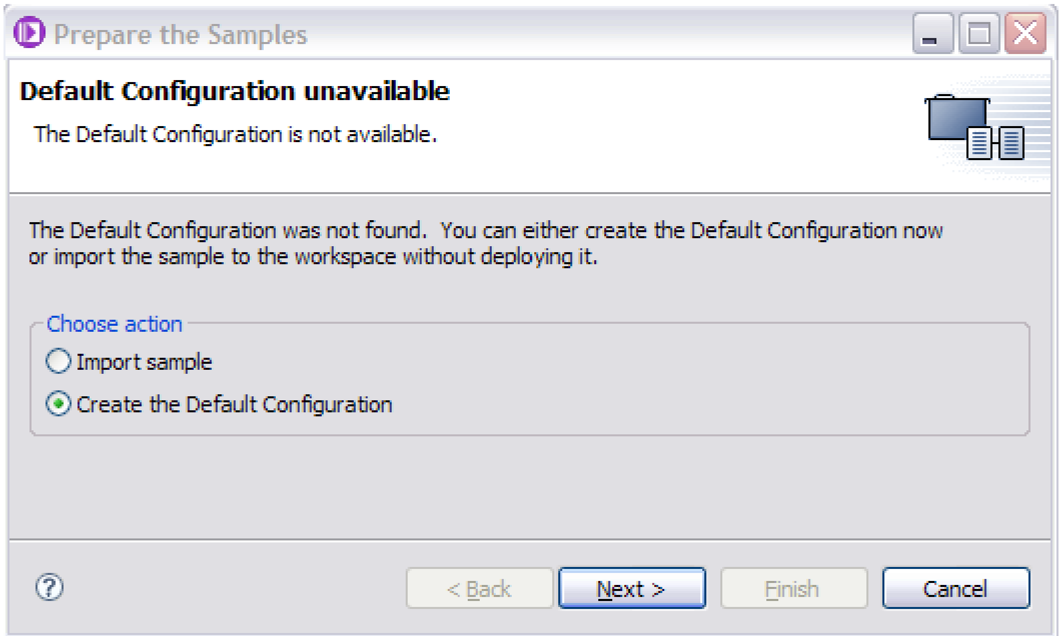
PHPCompute Node sample
This sample shows how to use the PHPCompute node in a message flow to transform an XML message.

XSL Transform sample
This sample shows how to use a message flow to transform an X XML message according to the rules provided by an XSL styleshe

____ *9.* Select the **JavaCompute Node sample**.

____ *10.* The WebSphere Message Broker Toolkit information center should now be presented to you on the page with the JavaCompute Node sample. You can read the information on this page and in the links. When ready, click the "**Import and deploy the sample**" link on the information center page.

____ *11.* If the broker does not have a default configuration available, a wizard prompt is displayed to create a default configuration.


Prepare the Samples

Default Configuration unavailable

The Default Configuration is not available.

The Default Configuration was not found. You can either create the Default Configuration now or import the sample to the workspace without deploying it.

Choose action

○ Import sample

● Create the Default Configuration

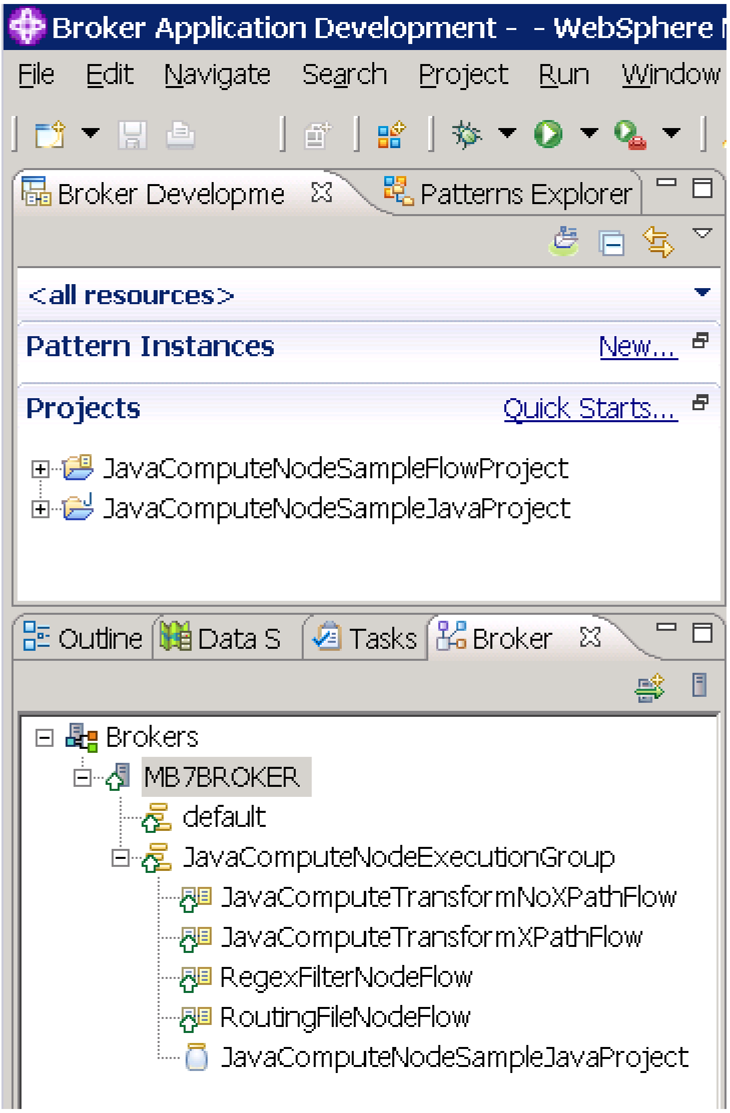
__ a. Click **Next**.

__ b. Another wizard prompt will be displayed with a **Welcome** screen summarizing the default configuration. Click **Next**.

__ *c.* Another Wizard prompt will be displayed with a **Default Configuration progress** screen. Click **Next**.

__ *d.* The wizard will go through the process of creating all the WebSphere MQ and WebSphere Message Broker artifacts required for this sample. It will then give you a summary; at this point click **Finish**.

__ e. You should now see the MB7BROKER broker in the Broker tab in the bottom left view of the Toolkit, and you should see two projects in the Broker Development view in the top left part of the Toolkit, as follows. If you do not see the two projects, then go back to the sample in the information center, select the **JavaCompute Node sample**, and once again click the "**Import and deploy the sample**" link.


Broker Application Development - - WebSphere
File Edit Navigate Search Project Run Window
Broker Developme
Patterns Explorer
<all resources>
Pattern Instances
New...
Projects
Quick Starts...
JavaComputeNodeSampleFlowProject
JavaComputeNodeSampleJavaProject
Outline
Data S
Tasks
Broker
Brokers
MB7BROKER
default
JavaComputeNodeExecutionGroup
JavaComputeTransformNoXPathFlow
JavaComputeTransformXPathFlow
RegexFilterNodeFlow
RoutingFileNodeFlow
JavaComputeNodeSampleJavaProject

____ 12. Test the message flow to ensure the vanilla sample is working. Use the instructions in the **Running the Sample** link on the information center page from step 10 above (____ 10).

____ 13. Now you are ready to prepare the Broker environment for accessing XC10 data grids and to modify this sample JavaCompute node to access a data grid. (In the WebSphere Message Broker Toolkit) update the Java project with the appropriate ObjectGrid library. NOTE this likely will be the same library (with the same requirement for the standalone version of the WebSphere eXtreme Scale client) used in step 4 above (____ 4).

__ a. Right-click **JavaComputeNodeSampleJavaProject**, and then select **Properties.**

__ b. On the **Properties** popup window, select **Java Build Path**.

__ c. Select the **Libraries** tab, and select **Add External JARS**.

__ d. You will need to add the WebSphere eXtreme Scale ogclient.jar file here (the wsogclient.jar will NOT work for this configuration). If you do not have this jar file available, see step 4 above (____ 4) to install the stand-alone version of the WebSphere eXtreme Scale client. When you have the client, navigate to your >**root**<eXtremeScale\ObjectGrid\lib directory and select the **ogclient.jar** file.

__ e. Select **OK**.

____ 14. Configure the Broker class loader to load the above same ObjectGrid library. See the WebSphere Message Broker V7 information center page ac37125_ for more information about the options available for this step. For purposes of this demonstration, add the ObjectGrid library to the Windows class path.

__ a. In Windows, go to **Start**, **Control Panel**, and select **System**.

__ b. Select the **Advanced** tab, and then select **Environment Variables**.

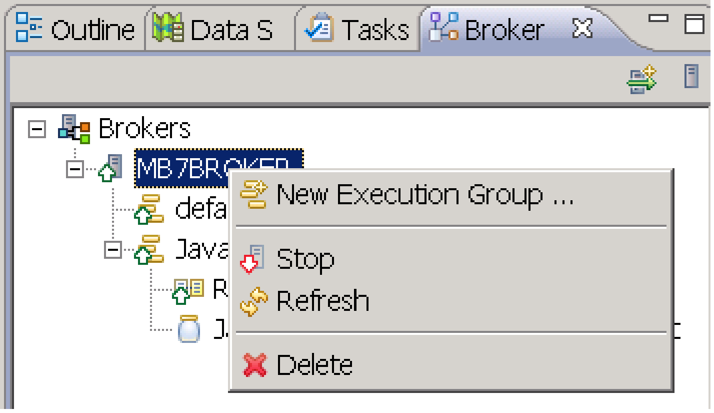
__ c. Edit the **CLASSPATH** and add the full library name for either the ogclient.jar as in #12 above. **NOTE** see APAR IC78067 if you have issues running the WebSphere MQ **mqsilist** command with ogclient.jar in the Class path.

__ d. Select **OK** multiple times to get all the way out of **System**.

__ e. Confirm the change by starting the Windows command console and enter **SET CLASSPATH**.

____ 15. Restart the Broker.

__ a. (In the WebSphere Message Broker Toolkit) in the bottom left pane, right-click MB7BROKER, and select Stop.



__ b. Once the Broker is stopped (red down arrow precedes the Broker name), right-click MB7BROKER, and select Start. Wait for the Broker to start (green up arrow precedes the Broker name).

____ 16. (In the WebSphere Message Broker Toolkit) prepare the **JavaCompute** node for editing:

__ a. Expand the **JavaComputeNodeSampleFlowProject.**

__ b. Fully expand **Flows.**

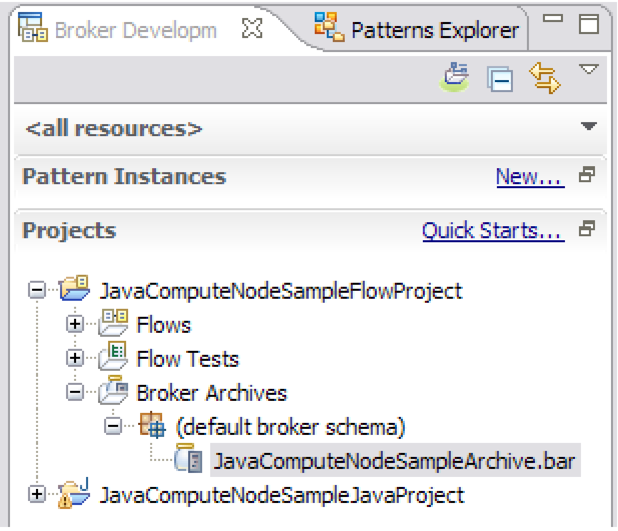
__ c. Right click **RegexFilterNodeFlow.msgflow**, and select **Open**.

__ d. On the message flow presented in the edit view, right-click the **RegexFilterNode** and select **Open Java**. You should now see the Java code associated with this **JavaCompute** node.

____ 17. Add the Java code to access the XC10 data grid to the JavaCompute node; Appendix B has the final sample code with the additions for accessing XC10 data grids. There are 3 sections of the original code from the SimpleGridDemo program in Appendix A that will be inserted as is into the sample code for the JavaCompute node. In both appendices, there are comments denoting where the sections begin and end.

____ 18. Deploy the modified sample message flow.

__ a. In the upper-left pane, under the Flow project (not the Java project), double click **JavaComputeNodeSampleArchive.bar**.



Broker Developm | Patterns Explorer

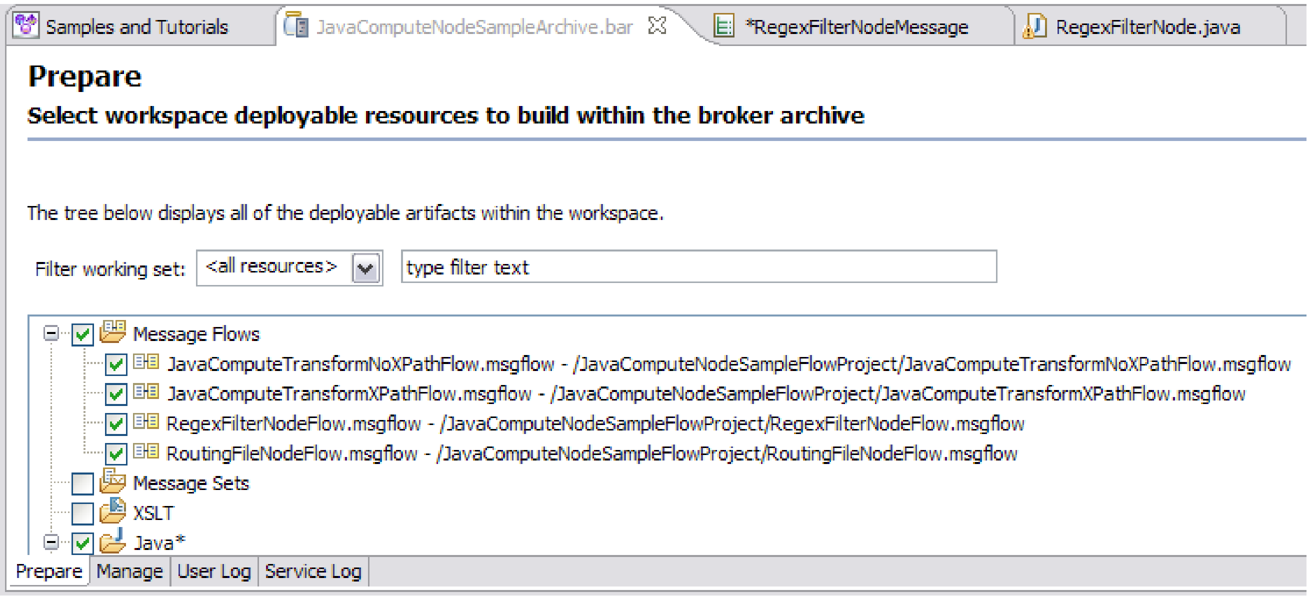
<all resources>

Pattern Instances New...

Projects Quick Starts...

- JavaComputeNodeSampleFlowProject
  - Flows
  - Flow Tests
  - Broker Archives
    - (default broker schema)
      - JavaComputeNodeSampleArchive.bar
- JavaComputeNodeSampleJavaProject

____ 19. This will present a new view In the upper right screen; select the **Prepare** tab.



Samples and Tutorials | JavaComputeNodeSampleArchive.bar | *RegexFilterNodeMessage | RegexFilterNode.java

Prepare

Select workspace deployable resources to build within the broker archive

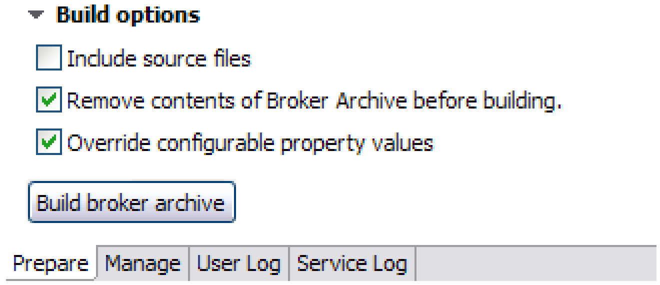
The tree below displays all of the deployable artifacts within the workspace.

Filter working set: <all resources> type filter text

- ☑ Message Flows
  - ☑ JavaComputeTransformNoXPathFlow.msgflow - /JavaComputeNodeSampleFlowProject/JavaComputeTransformNoXPathFlow.msgflow
  - ☑ JavaComputeTransformXPathFlow.msgflow - /JavaComputeNodeSampleFlowProject/JavaComputeTransformXPathFlow.msgflow
  - ☑ RegexFilterNodeFlow.msgflow - /JavaComputeNodeSampleFlowProject/RegexFilterNodeFlow.msgflow
  - ☑ RoutingFileNodeFlow.msgflow - /JavaComputeNodeSampleFlowProject/RoutingFileNodeFlow.msgflow
- ☐ Message Sets
- ☐ XSLT
- ☑ Java*

Prepare | Manage | User Log | Service Log

__ a. Let all selections default. Scroll to the bottom of this pane, and click the **Build broker archive** button.



Build options

□ Include source files

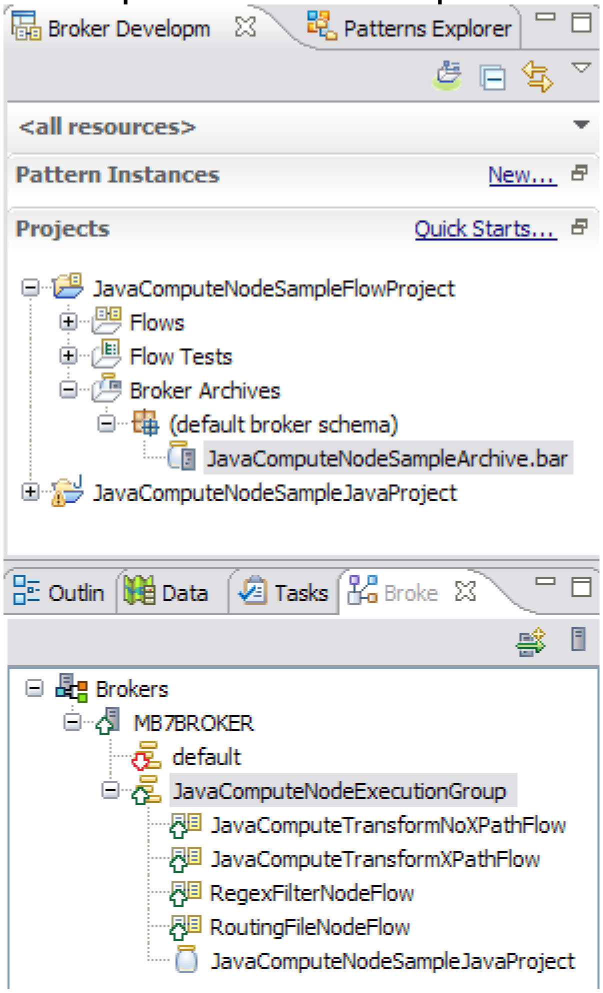
☑ Remove contents of Broker Archive before building.

☑ Override configurable property values

__ b. If you get a **Save Changes?** pop-up, select **Yes**.

__ c. At the **Operation Completed Successfully** popup, select **OK** (or if unsuccessful, click the **Details** to get more info about the issue).

__ d. In the upper left pane, click and drag **JavaComputeNodeSampleArchive.bar** down to **JavaComputeNodeExecutionGroup** in the lower left pane.



__ e. If you get a **Save Changes?** popup, select **Yes**.

__ f. A popup **progress** window will be displayed, and if the deploy is successful, the popup will disappear. If the deploy is not successful, click **Details** for more information about the issue.

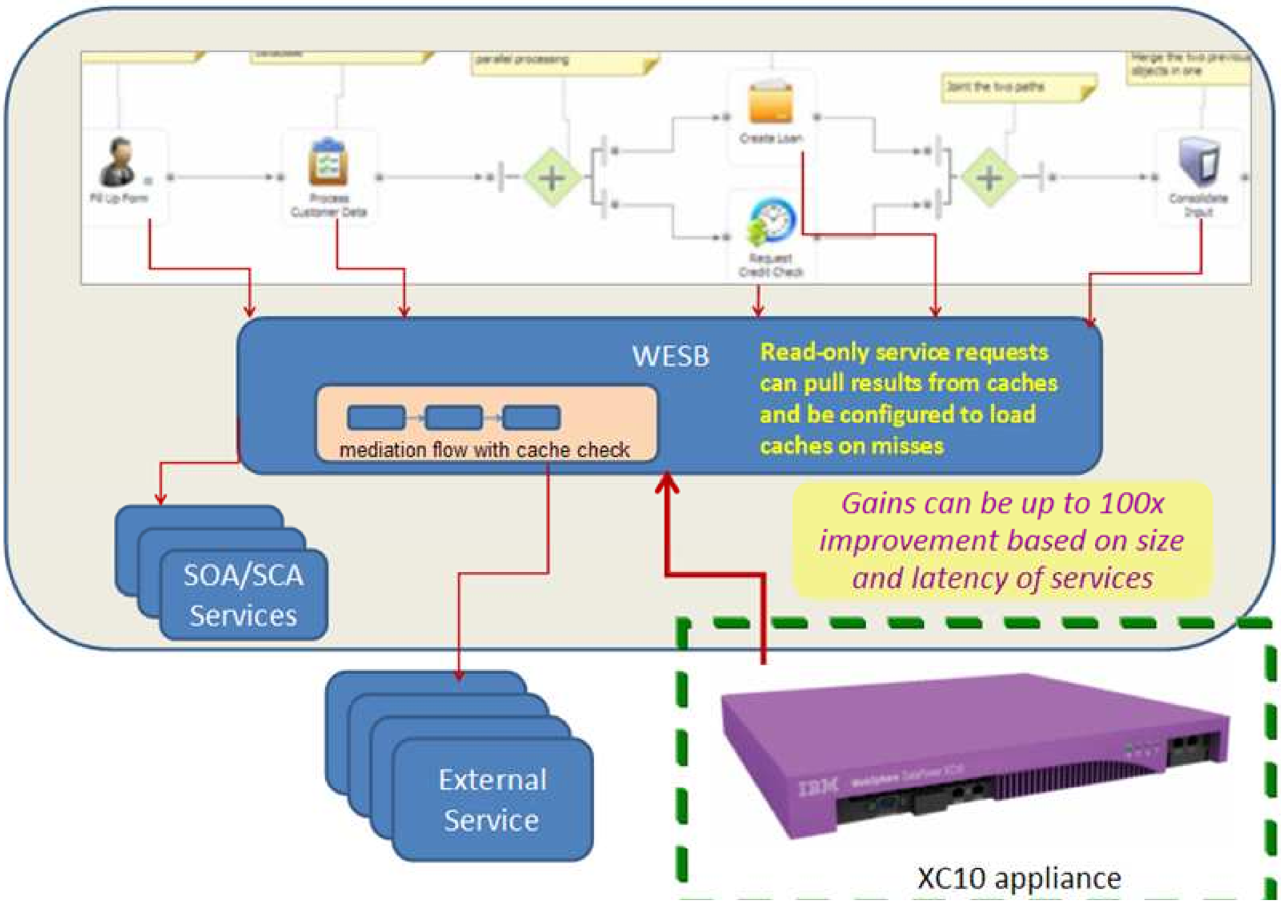
__ g. You should now be ready to run the modified sample in exactly the same way you did in step 12 above (____ 12).

__ h. In addition to verifying results as with the vanilla sample, also look in C:\Documents and Settings\All Users\Application Data\IBM\MQSI\components\MB7BROKER\<execution group id>\console.txt for the WebSphere eXtreme Scale messages, any stack traces you may have, and the results of the System.out.println commands from the Java program. See **Appendix C** for a sample result of console.txt after running the message flow as-is a second time and thus attempting to insert a duplicate key into the data grid.

Part 3: Accessing an XC10 data grid from WebSphere Enterprise Service Bus

In Part 3 of this lab, you will take an existing Integration Designer sample for a WebSphere Enterprise Service Bus server, and modify it to access a DataPower XC10 data grid. The Stock Quote sample will be the basis for this lab.

WebSphere Process Server or WebSphere Enterprise Service Bus caching with DataPower XC10

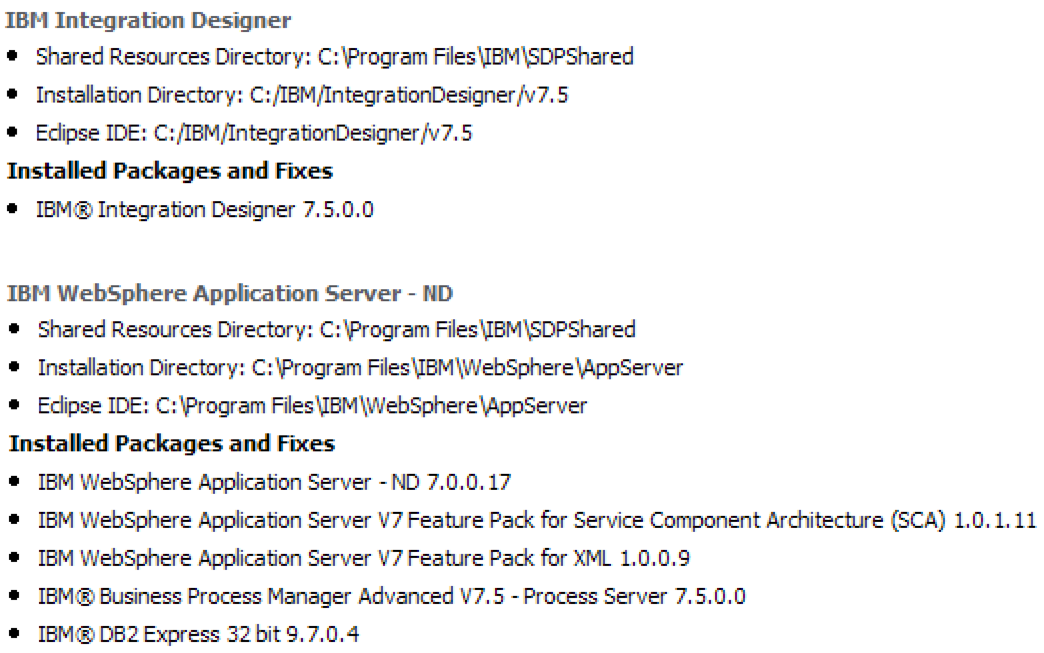

parallel processing
Joint the two paths
Merge the two previous objects in one
Fill Up Form
Process Customer Data
Create Loan
Request Credit Check
Consolidate Input
WESB
Read-only service requests can pull results from caches and be configured to load caches on misses
mediation flow with cache check
Gains can be up to 100x improvement based on size and latency of services
SOA/SCA Services
External Service
XC10 appliance

Note that for purposes of this lab, the calls made to the XC10 data grid have little relationship to the existing Stock Quote sample; they demonstrate how the data grid can be accessed for retrievals and inserts.

NOTE: Caching within WebSphere Enterprise Service Bus mediation flows implies that the data being cached is from web services responses, and thus you can only retrieve objects from the data grid that are inserted by the cache insert mediator. If objects in the grid are inserted by anything other then the ObjectGrid insert (for example, a Java stand-alone program) the ObjectGrid retrieve will fail with an exception.

This lab was run with IBM Integration Designer V7.5 and BPM V7.5, although the V7.0 products should work the same. Only the WebSphere Enterprise Service Bus test environment is required to be installed for this lab.

The following are the product details from IBM's Installation Manager used to build this lab:



IBM Integration Designer

- Shared Resources Directory: C:\Program Files\IBM\SDPShared
- Installation Directory: C:/IBM/IntegrationDesigner/v7.5
- Eclipse IDE: C:/IBM/IntegrationDesigner/v7.5

Installed Packages and Fixes

- IBM® Integration Designer 7.5.0.0

IBM WebSphere Application Server - ND

- Shared Resources Directory: C:\Program Files\IBM\SDPShared
- Installation Directory: C:\Program Files\IBM\WebSphere\AppServer
- Eclipse IDE: C:\Program Files\IBM\WebSphere\AppServer

Installed Packages and Fixes

- IBM WebSphere Application Server - ND 7.0.0.17
- IBM WebSphere Application Server V7 Feature Pack for Service Component Architecture (SCA) 1.0.1.11
- IBM WebSphere Application Server V7 Feature Pack for XML 1.0.0.9
- IBM® Business Process Manager Advanced V7.5 - Process Server 7.5.0.0
- IBM® DB2 Express 32 bit 9.7.0.4

Details to create/import and run the **Stock Quote** sample as-is are not provided in this lab, as they are already provided in the Integration Designer information center. To get to this information, start the **IBM Integration Designer** with a new workspace, go to the IBM Integration Designer samples gallery, and you should see the Stock Quote sample. There are options to **Import** it and to **View Instructions**. If you are not familiar with the IBM Integration Designer and running WebSphere Enterprise Service Bus applications in the integrated test client, then you may want to first go through the exercise of running this sample as-is and become familiar with it before making the changes required to integrate it with the XC10.

This lab will focus on the steps required after the **Stock Quote** sample has been created or imported into the **IBM Integration Designer**.

____ 20. Install the required XC10 mediation primitives into IBM Integration Designer.

NOTE: At the time of writing, these mediation primitives only come packaged with the full server installation of WebSphere eXtreme Scale (versus the client installation), either in a stand-alone installation or in a WebSphere Application Server installation.

__ a. The mediation primitive plug-in jar file is called: com.ibm.bpm.OGCache_1.0.0.jar, found either in:

1) the <install_root>/ObjectGrid/wesb directory for a stand-alone installation of WebSphere eXtreme Scale, or

2) the <was_home>/optionalLibraries/ObjectGrid/wesb directory for a WebSphere Application Server install of WebSphere eXtreme Scale.

__ b. Copy the OGCache jar file into the Integration Designer plugins directory, in this lab environment's case to C:\IBM\IntegrationDesigner\v7.5\plugins.

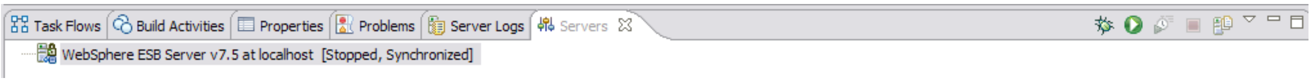
____ 21. Restart the Integration Designer with the "- clean" parameter by going to the directory C:\IBM\IntegrationDesigner\v7.5 in a command screen and executing "wid.exe –clean".

____ 22. Install the required ObjectGrid mediation primitives into WebSphere Enterprise Service Bus.

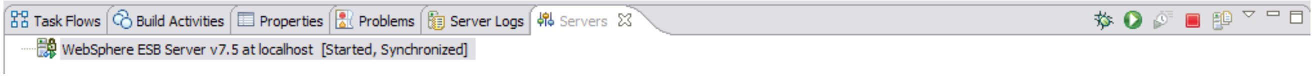
__ a. Copy the wsogclient.jar file from your WebSphere eXtreme Scale installation (server or client) into <was_home>/lib/ext, in this lab environment's case to C:\Program Files\IBM\WebSphere\AppServer\lib\ext.

__ b. Copy the GetFromCache.jar file from the same directory as step 20_a above (____ 20.__ a), also into <was_home>/lib/ext.

____ 23. Restart the WebSphere Enterprise Service Bus server in the Integration Designer by going to the Server view, and whether the server is already started or stopped, click the green right arrow to restart the server.



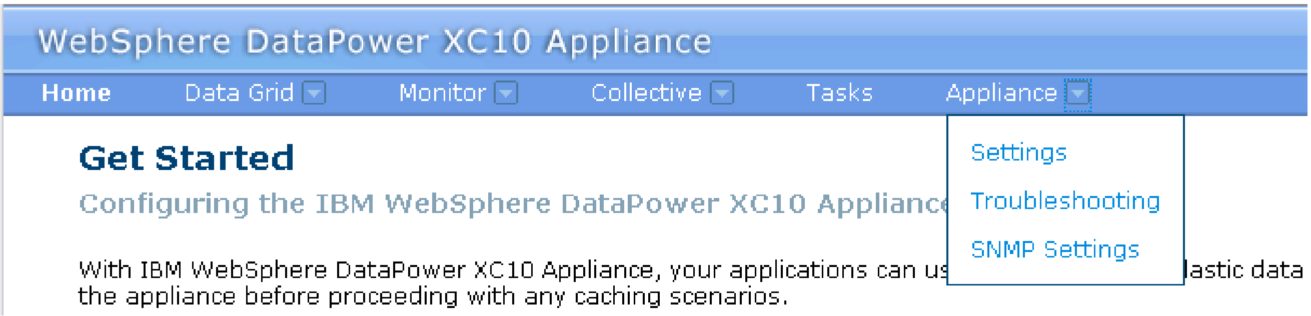
WebSphere ESB Server v7.5 at localhost [Stopped, Synchronized]



WebSphere ESB Server v7.5 at localhost [Started, Synchronized]

____ 24. For this lab's testing purposes, you will "disable" any TLS requirements for communication between the WebSphere Enterprise Service Bus server and the XC10 appliance. First, ensure the XC10 appliance has the TLS requirement disabled.

__ a. In the XC10 web console, select the **Appliance** down arrow, and select **Settings**.

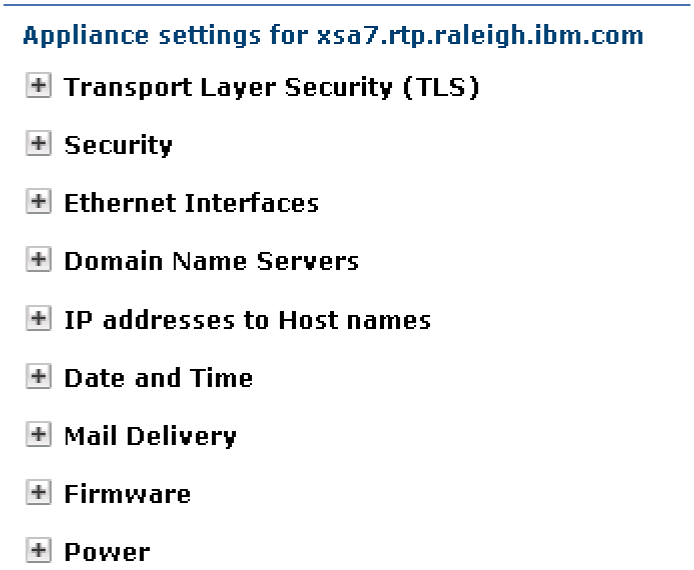


Get Started

Configuring the IBM WebSphere DataPower XC10 Appliance

With IBM WebSphere DataPower XC10 Appliance, your applications can us lastic data the appliance before proceeding with any caching scenarios.

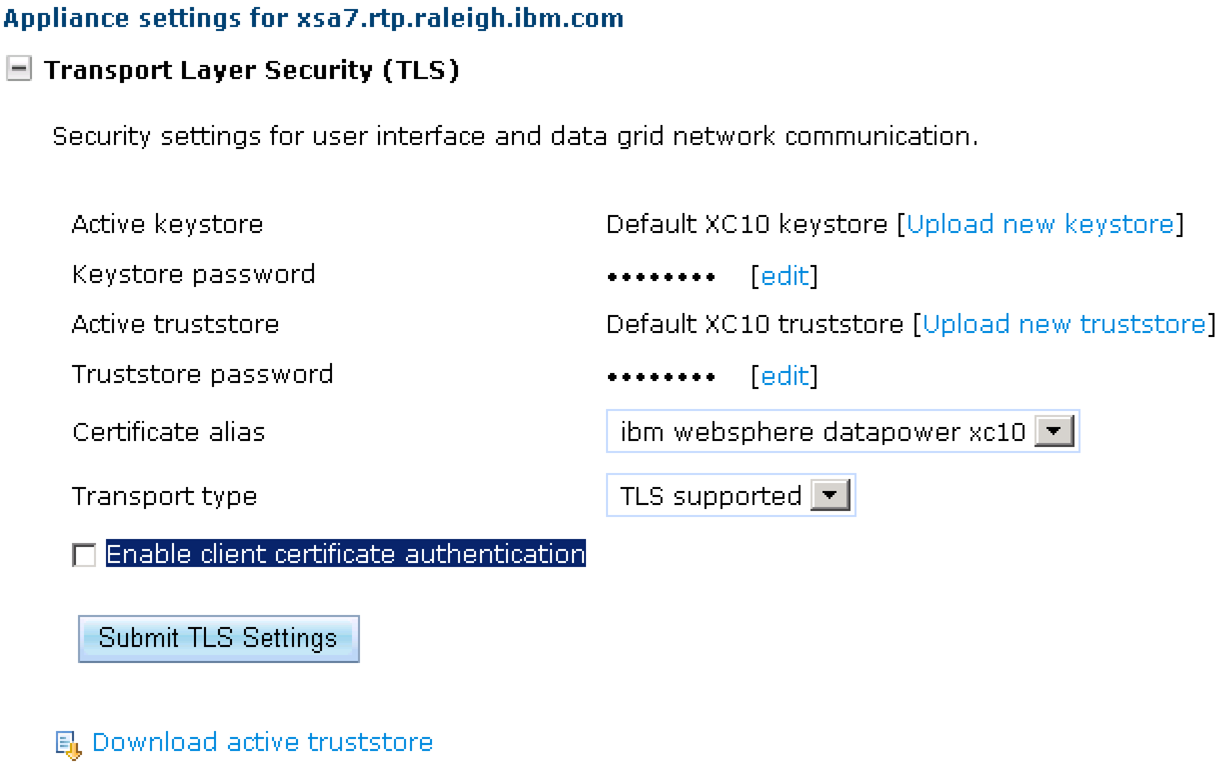
__ b. Expand the **Transport Layer Security (TLS)** option.



Appliance settings for xsa7.rtp.raleigh.ibm.com

- ⊞ Transport Layer Security (TLS)
- ⊞ Security
- ⊞ Ethernet Interfaces
- ⊞ Domain Name Servers
- ⊞ IP addresses to Host names
- ⊞ Date and Time
- ⊞ Mail Delivery
- ⊞ Firmware
- ⊞ Power

__ c. Ensure that the **Enable client certificate authentication** option is not selected.



Appliance settings for xsa7.rtp.raleigh.ibm.com

⊟ Transport Layer Security (TLS)

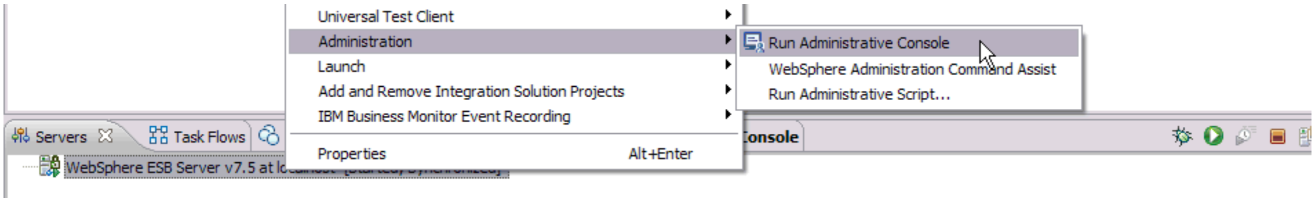
Security settings for user interface and data grid network communication.

Active keystore	Default XC10 keystore [Upload new keystore]
Keystore password	•••••••• [edit]
Active truststore	Default XC10 truststore [Upload new truststore]
Truststore password	•••••••• [edit]
Certificate alias	ibm websphere datapower xc10 ▾
Transport type	TLS supported ▾

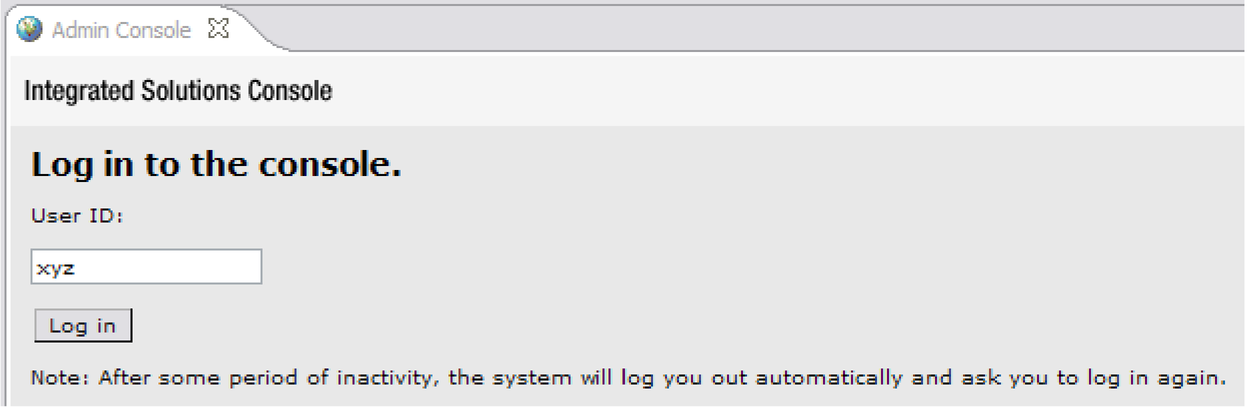
□ Enable client certificate authentication

____ 25. Now ensure the WebSphere Enterprise Service Bus server does not require security.

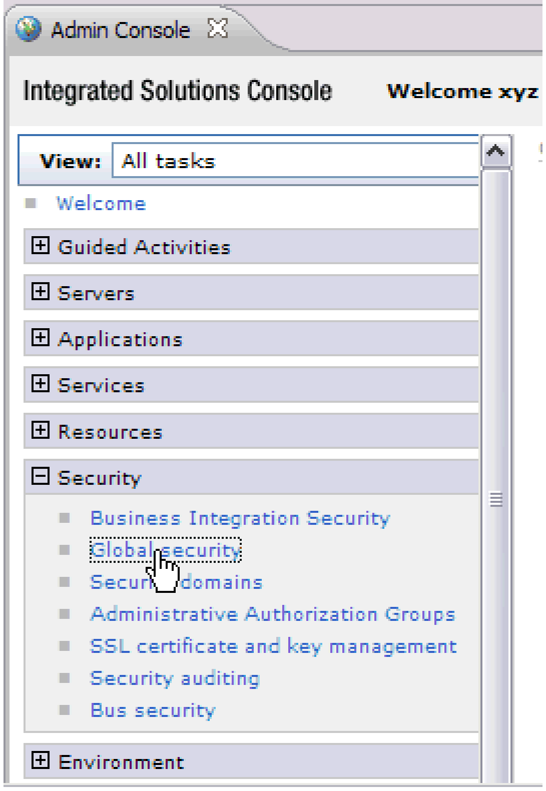
__ a. In the **IBM Integration Designer**, open the **administrative console** by ensuring the WebSphere Enterprise Service Bus server is running, right click on the WebSphere Enterprise Service Bus server, select **Administration**, and then select **Run Administrative Console**. (The **Run Administrative Console** option is not available if the server is not running.)


Universal Test Client
Administration
Launch
Add and Remove Integration Solution Projects
IBM Business Monitor Event Recording
Properties Alt+Enter
Run Administrative Console
WebSphere Administration Command Assist
Run Administrative Script...
Servers
Task Flows
Console
WebSphere ESB Server v7.5 at lo

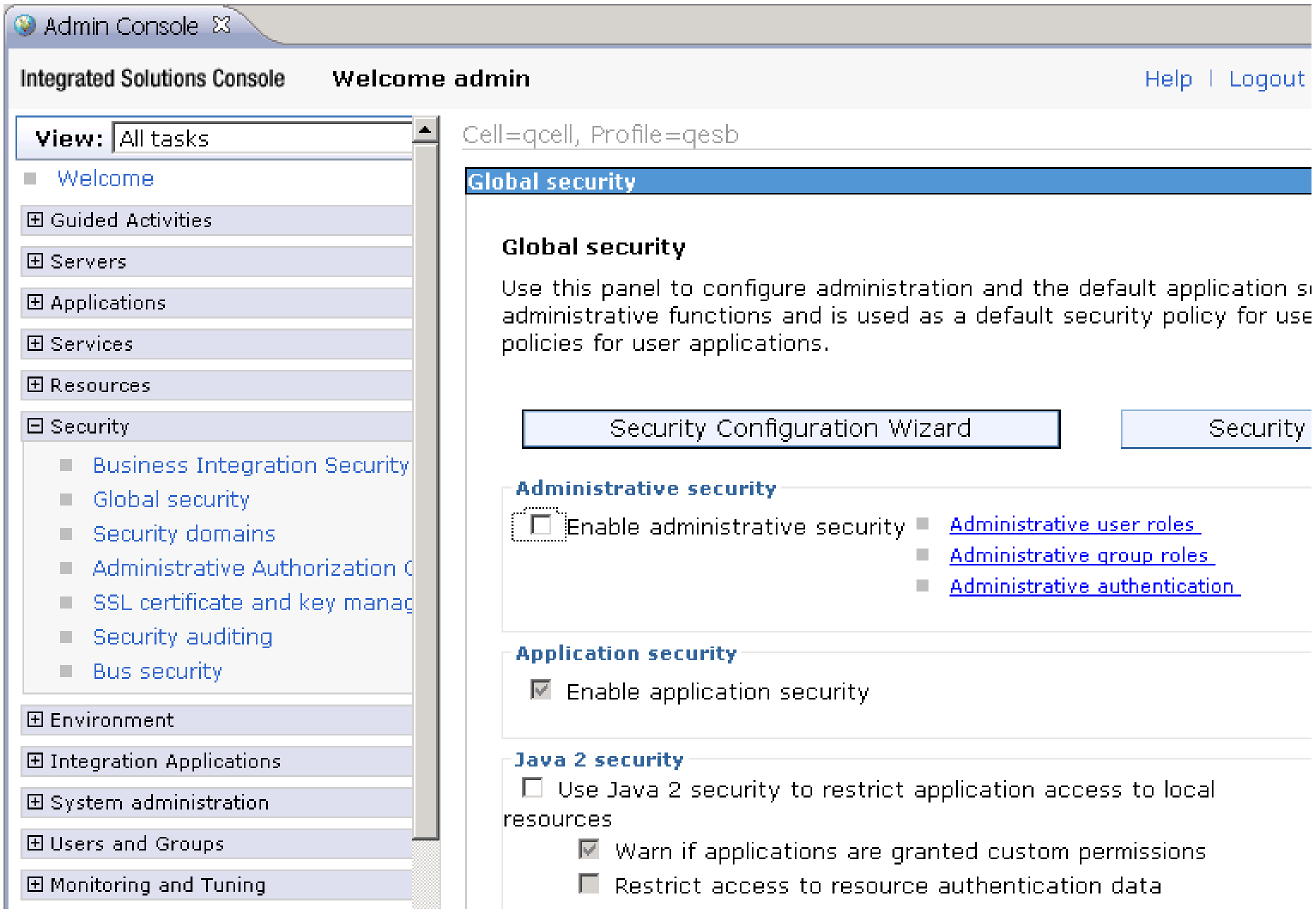
__ b. Log in to the console. Depending on how global (administrative) security is set up on the server, you may either log in with any ID you want (as below), or you may be prompted for a valid user ID / password. If the latter, use the default user ID / password of admin / admin.


Admin Console
Integrated Solutions Console
Log in to the console.
User ID:
xyz
Log in
Note: After some period of inactivity, the system will log you out automatically and ask you to log in again.

__ c. Expand the **Security** selection in the left pane, and select Global Security.



__ d. Ensure the **Enable administration security** is NOT selected (and thus **Enable application security** will be disabled also). Also ensure **Java 2 security** is not selected. If a change is made, then ensure the changes are **Applied** and **Saved**, and the WebSphere Enterprise Service Bus server is restarted as in step 23 above (____ 23).



Global security

Use this panel to configure administration and the default application s administrative functions and is used as a default security policy for use policies for user applications.

Administrative security

☐ Enable administrative security

Application security

☑ Enable application security

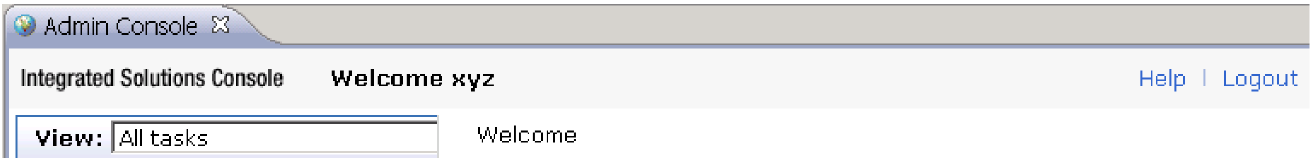
Java 2 security

☐ Use Java 2 security to restrict application access to local resources

☑ Warn if applications are granted custom permissions

☐ Restrict access to resource authentication data

__ e. Logout and close the administrative console.



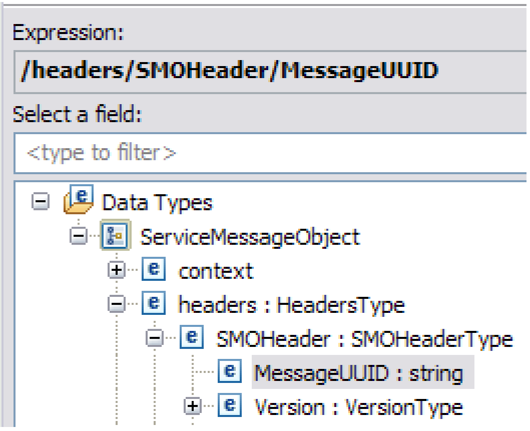
You are now ready to make the changes to the **Stock Quote** sample to use the ObjectGrid primitives.

The first step is to modify the **SubscriptionInformation** business object to include all data cached by the ObjectGrid primitives in the flow, in context form (to be available to both the Request and Response flows). For demonstration purposes in this lab, all five fields are inserted into and retrieved from the data grid and **symbol** will be used as the key.

NOTE: There are a couple of related considerations to think about when caching “WebSphere Enterprise Service Bus data” on an XC10 appliance:

- How long will the data remain in cache? If you use a data grid without a dynamic map name, the data will live forever, unless removed through some other mechanism. If you use a data grid with a dynamic map name, the default time-to-live of the data in cache is one hour, and at the time of writing, there is no means to adjust the time-to-live.

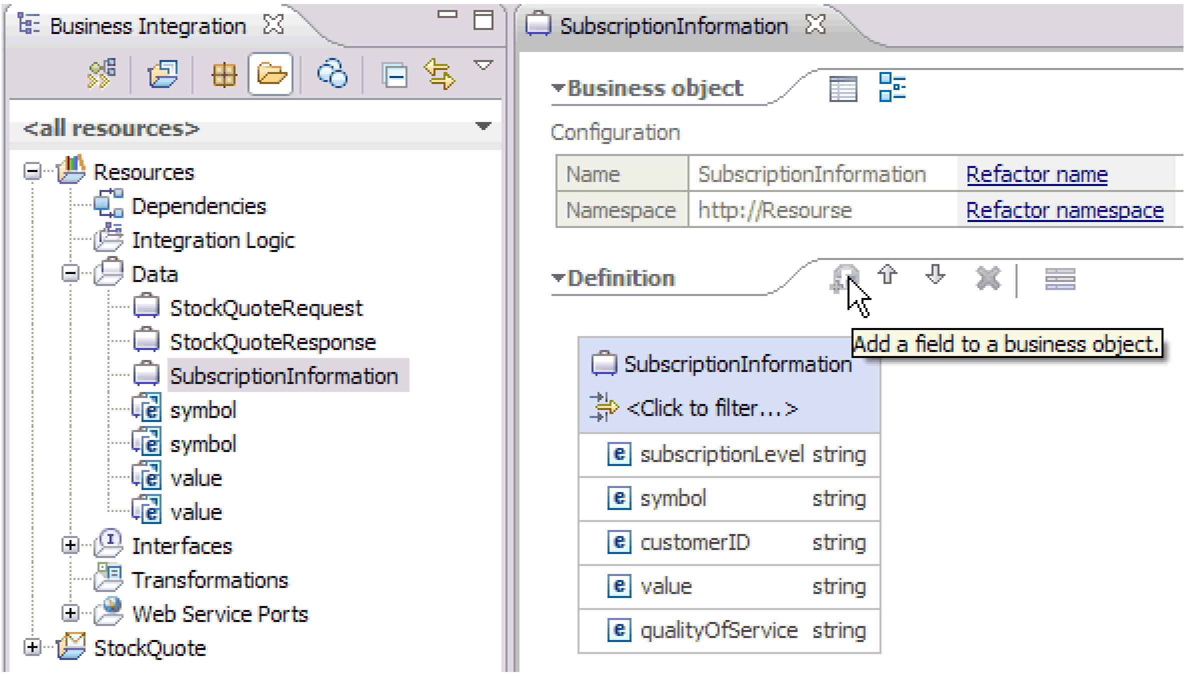
- If your mediation cannot tolerate a one hour time-to-live (or any specific value of a time-to-live, for that matter), then consider using the MesageUUID in the SMOHeader of the Service Message Object as the key to the data grid objects; see below.



Expression: **/headers/SMOHeader/MessageUUID**

Select a field: <type to filter>

____ 26. In the **Business Integration** view, expand **Resources**, then expand **Data**, and double-click **SubscriptionInformation**. Use the **add a field to a business object** button to add the 4 additional fields: **symbol**, **customerID**, **value,** and **qualityOfService**. All are **string**.



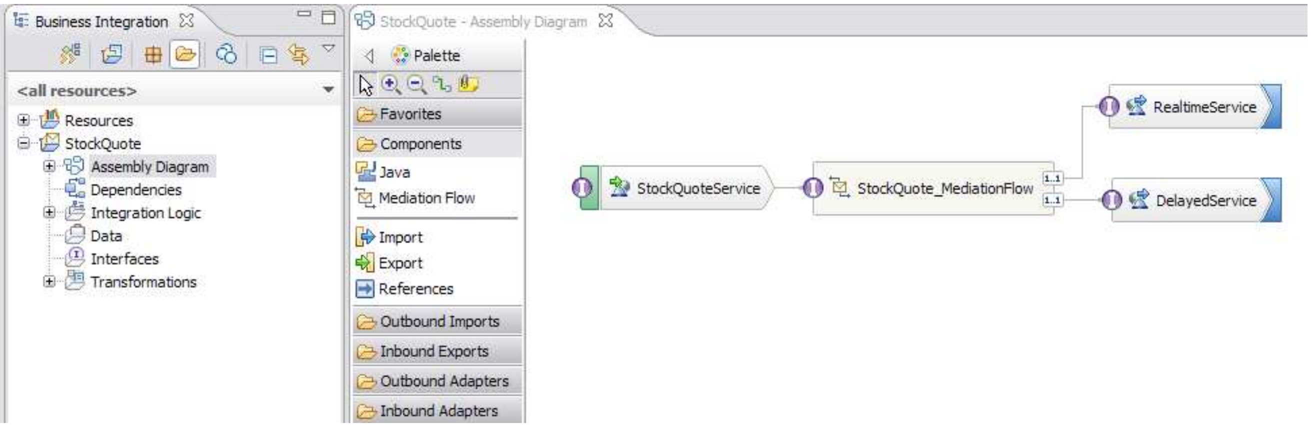
Name	SubscriptionInformation	Refactor name
Namespace	http://Resourse	Refactor namespace

SubscriptionInformation	
subscriptionLevel	string
symbol	string
customerID	string
value	string
qualityOfService	string

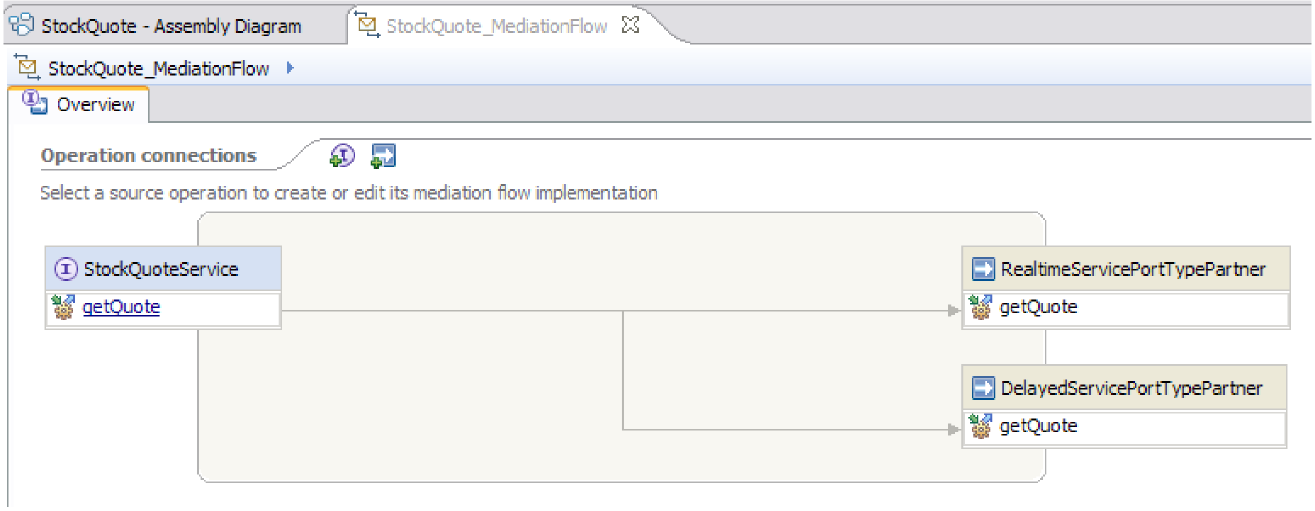
____ 27. Save the changes by going to **file**, and select **Save All**.

Now you will begin the process of adding and modifying the appropriate primitives in the existing **Stock Quote** mediation flow.

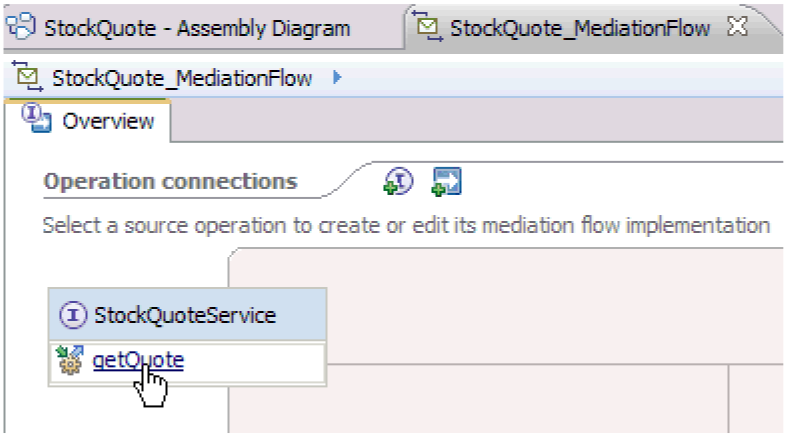
____ 28. In the **Business Integration** view, expand **StockQuote**, and then double-click **Assembly Diagram**. This will display StockQuote in the **Assembly Diagram editor**.



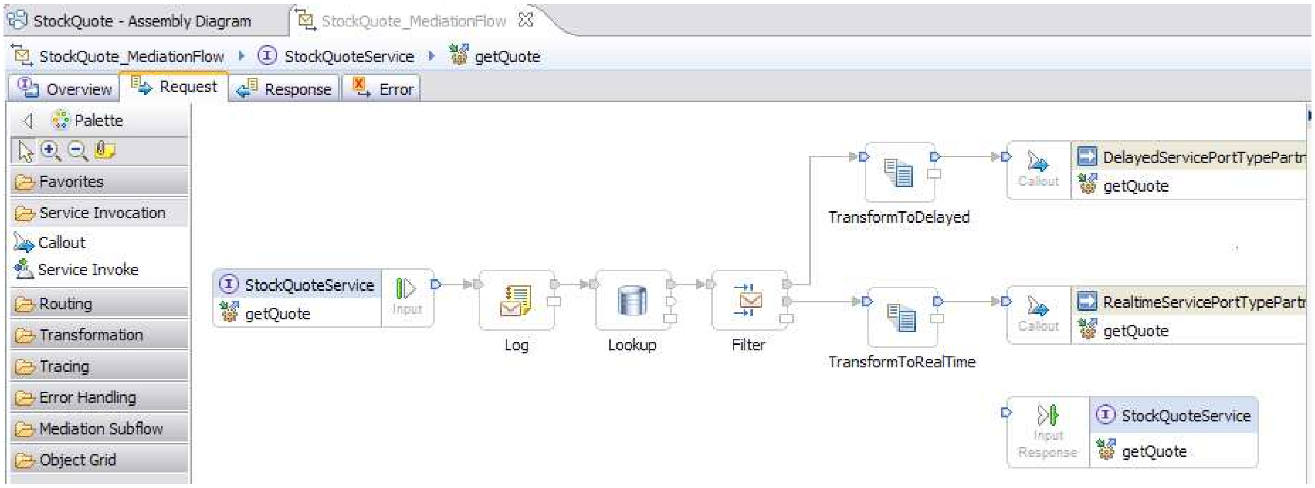
____ 29. In the editor panel, double-click **StockQuote_MediationFlow** to open the **Mediation Flow editor**.


StockQuote - Assembly Diagram
StockQuote_MediationFlow
StockQuote_MediationFlow
Overview
Operation connections
Select a source operation to create or edit its mediation flow implementation
StockQuoteService
getQuote
RealtimeServicePortTypePartner
getQuote
DelayedServicePortTypePartner
getQuote

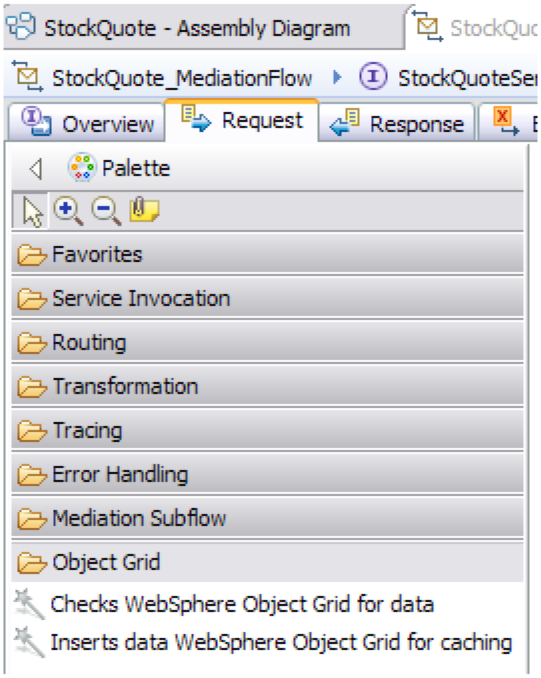
____ 30. Click the getQuote operation in the StockQuoteService interface to obtain the additional edit panels for the Mediation Flow.


StockQuote - Assembly Diagram
StockQuote_MediationFlow
StockQuote_MediationFlow
Overview
Operation connections
Select a source operation to create or edit its mediation flow implementation
StockQuoteService
getQuote

____ 31. As now seen, this will open additional editors, namely the **Request**, **Response** and **Error** editors. Click the **Request** tab to see the **Request** flow canvas, as shown below.



____ 32. Note above and below, this is where the **Object Grid** folder and primitives should be seen in the Palette. If they are not seen, then review the actions taken in steps 20 through 23 above (____ 20).

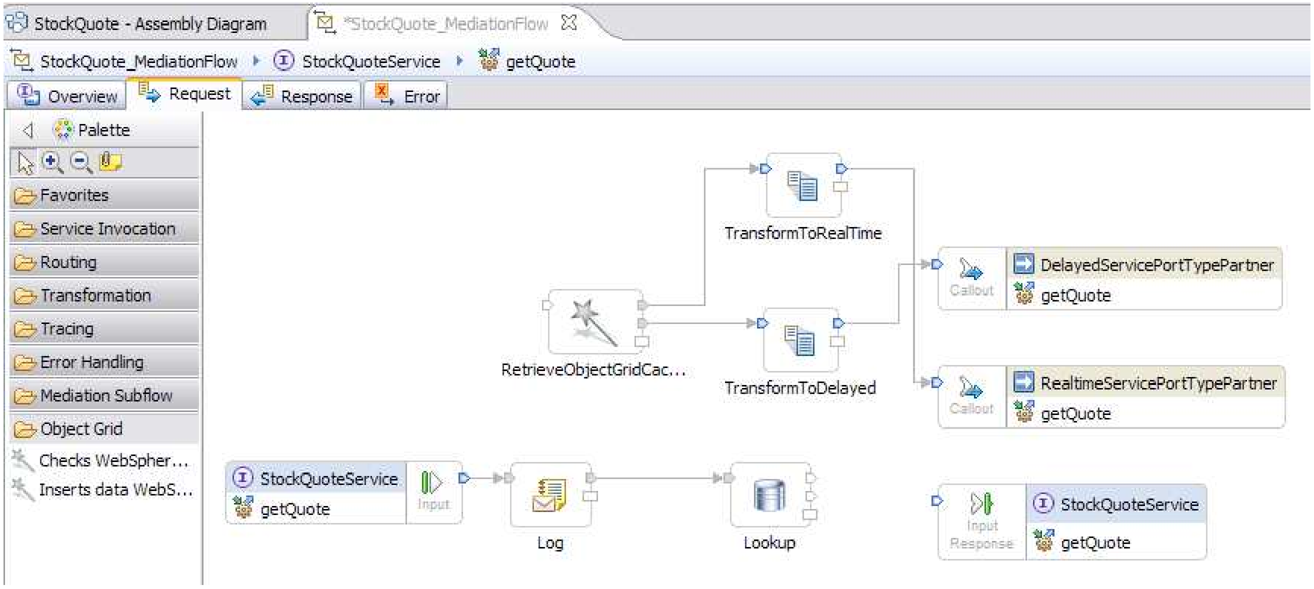


For demonstration purposes in this lab, on the Request flow you will replace the **Filter** primitive with the **Checks WebSphere Object Grid for data** primitive. Upon the ObjectGrid retrieval, if the data is found in the data grid, the flow will follow the “real-time” path; if the data is not found in the data grid, the flow will follow the “delayed” path.

____ 33. On the **Request** flow, right-click the Filter primitive, and select Delete.

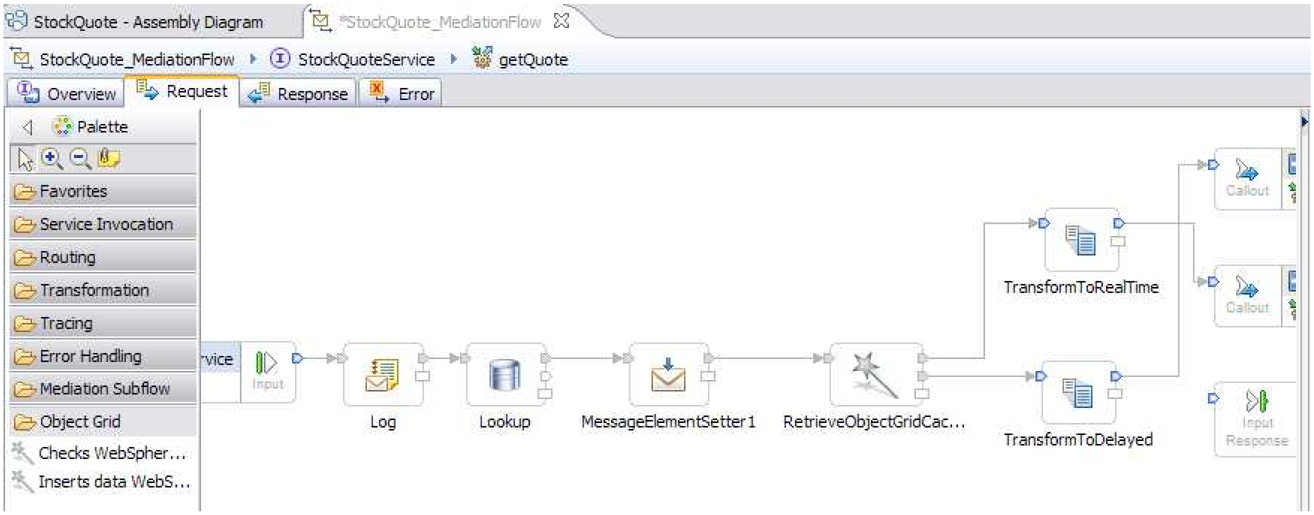
____ 34. Drag and drop the **Checks WebSphere Object Grid for data** primitive onto the **Request** flow canvas. Wire the **out** terminal of **RetrieveObjectGridCac…** to the **in** terminal of **TransformToRealTime**. Wire the **notInGrid** terminal of **RetrieveObjectGridCac…** to the **in** terminal of **TransformToDelayed**. Right click the canvas, and ensure **Automatic Layout** is selected.

NOTE: that at the time of writing with the available version of com.ibm.bpm.OGCache_1.0.0.jar, the **fail** terminal is not used at all in **RetrieveObjectGridCac…**

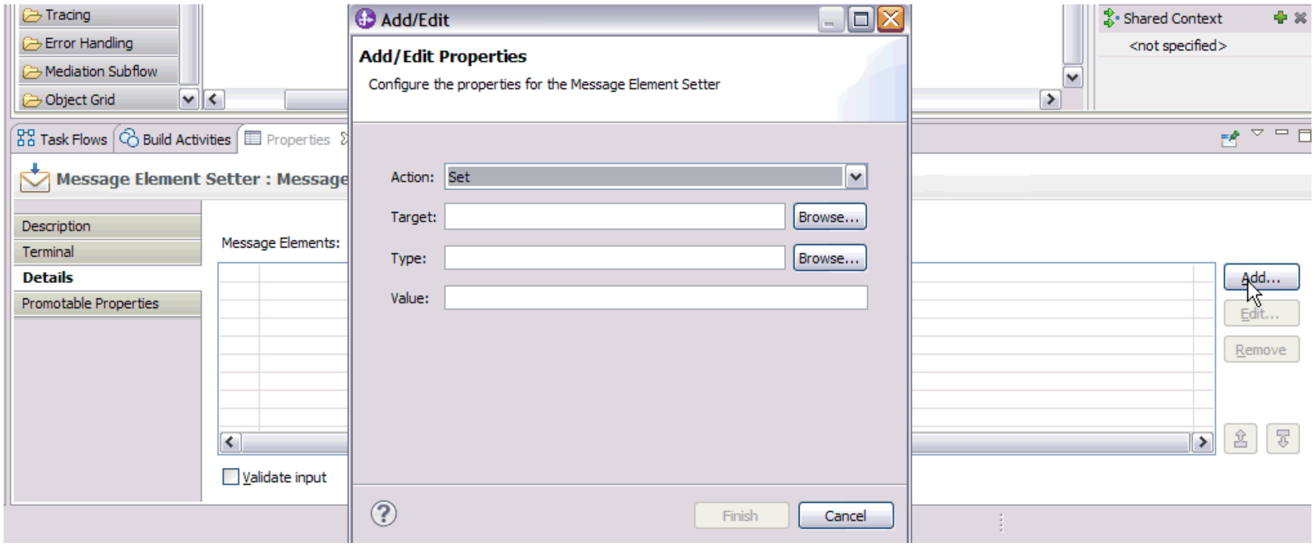


You now need to add a **Message Element Setter** primitive to copy the fields into the **SubscriptionInformation** business object from the b**ody** into the **context correlation** to make the fields available to both the **Request** and **Response** flows. On this **Request** flow, the 2 fields to be copied are **customerID** and **symbol**.

____ 35. Drag and drop the **Message Element Setter** primitive onto the **Request** flow canvas. Wire the **out** terminal of **Lookup** to the **in** terminal of **MessageElementSetter1**. Wire the **out** terminal of **MessageElementSetter1** to the **in** terminal of **RetrieveObjectGridCac…**



____ 36. Double-click the **MessageElementSetter1** primitive, and go to the **Properties** view below the canvas. In the **Properties** view, click **Details** on the left. On the right, click the **Add** button to **Add/Edit Properties**.



Add/Edit Properties

Configure the properties for the Message Element Setter

Action: Set

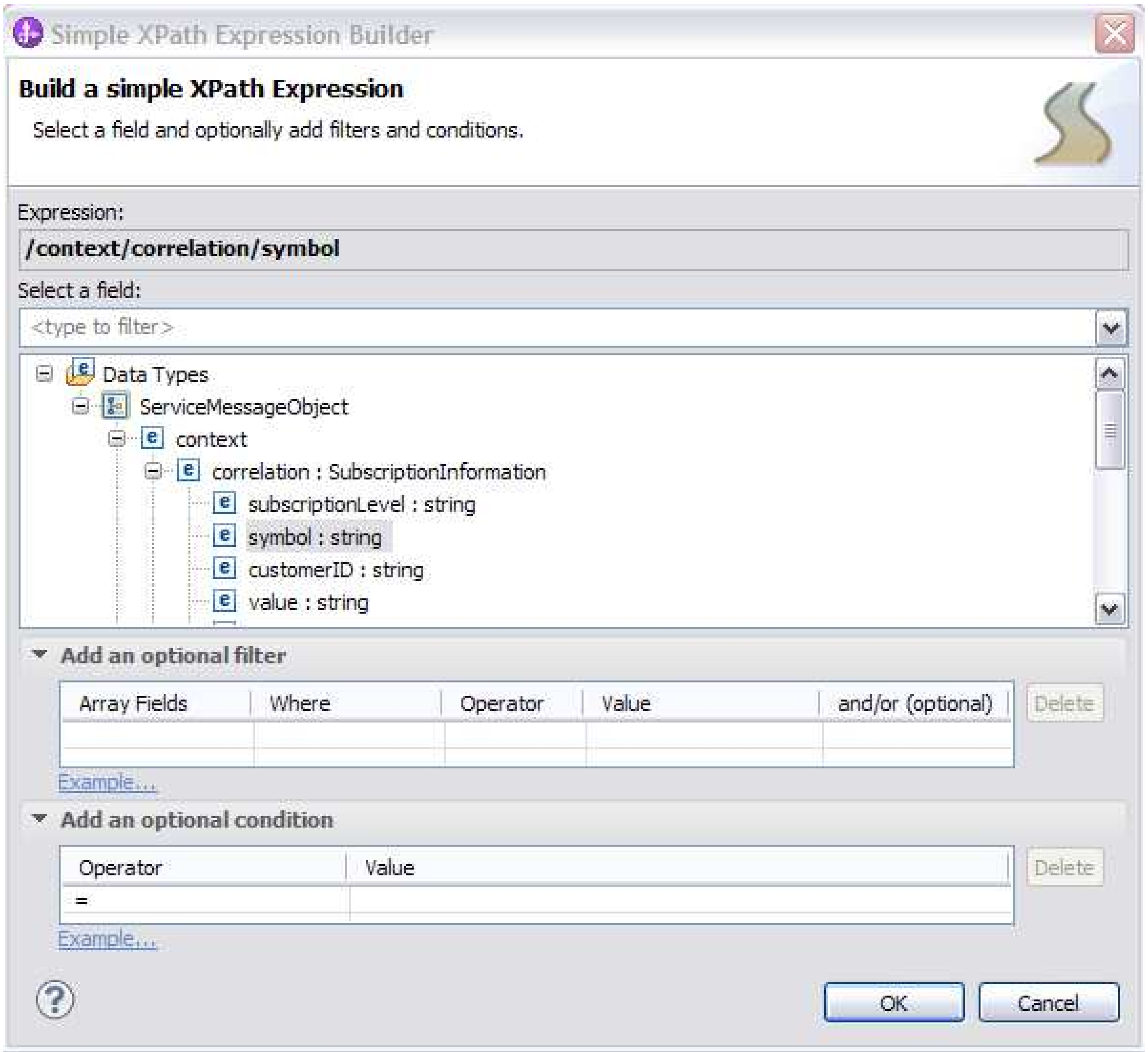
Target:

Type:

Value:

____ 37. On the **Action** pulldown, select **Copy**. You will copy the 2 fields from the **body** of the **StockQuoteRequest** business object to the **context correlation** of **SubscriptionInformation**.

__ a. For the **Target**, click **Browse** to build the XPath expression. Expand the **context** part of the path to get to **symbol**, and select OK.



Simple XPath Expression Builder

Build a simple XPath Expression

Select a field and optionally add filters and conditions.

Expression: **/context/correlation/symbol**

Select a field:

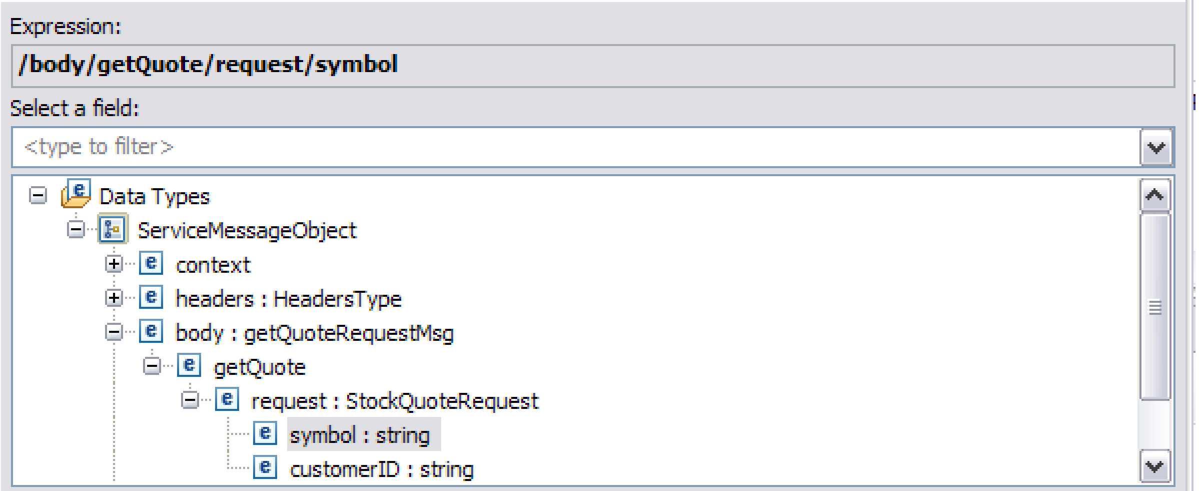
Add an optional filter

Array Fields	Where	Operator	Value	and/or (optional)

Add an optional condition

Operator	Value
=	

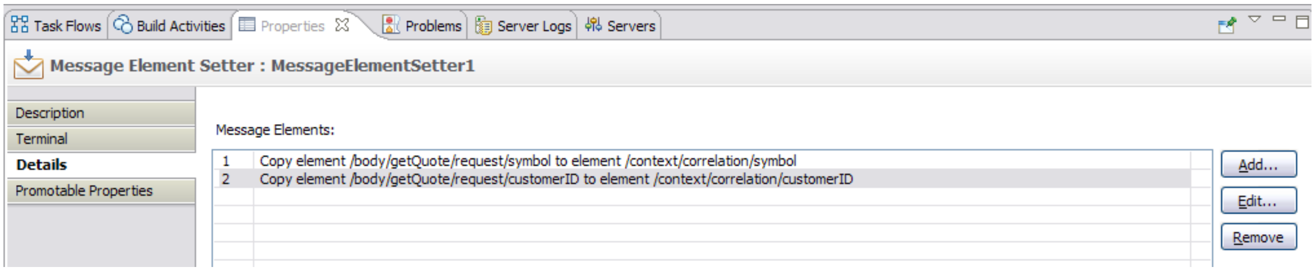
__ b. Similarly, for the **Source**, click **Browse**, expand the **body** portion of the path, select **symbol**, and OK and then **Finish**.



Expression: **/body/getQuote/request/symbol**

Select a field:

__ c. Repeat the steps to **Add** another **copy** element for the **customerID** field. You should now have the following **Detail** properties for **MessageElementSetter1.**

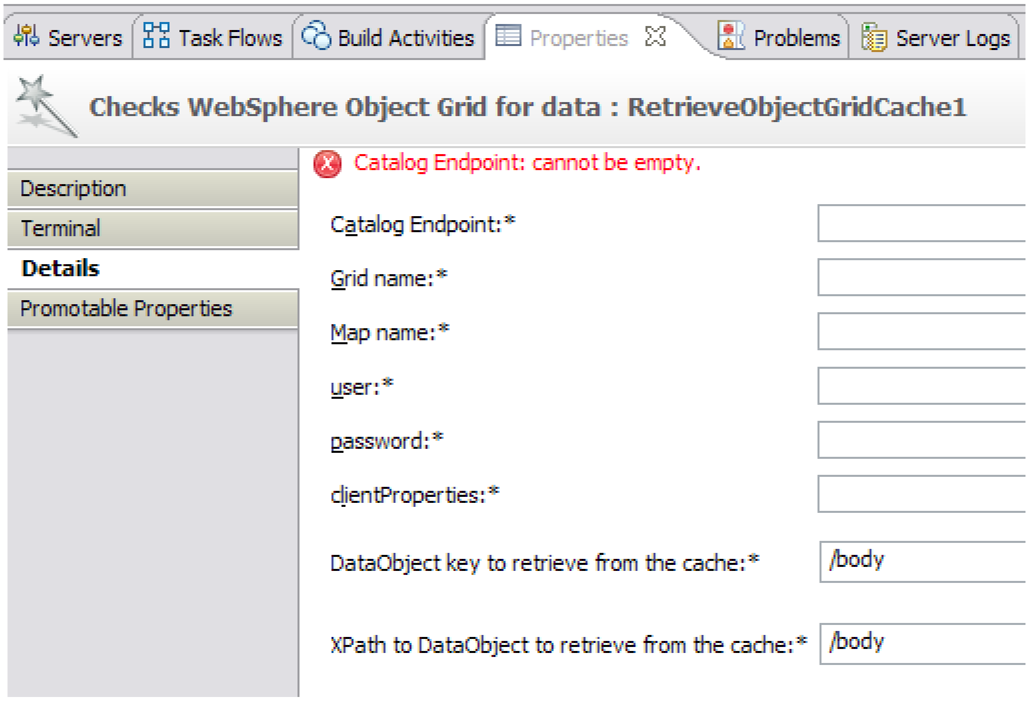


Message Element Setter : MessageElementSetter1

Message Elements:

1	Copy element /body/getQuote/request/symbol to element /context/correlation/symbol
2	Copy element /body/getQuote/request/customerID to element /context/correlation/customerID

____ 38. Double-click **RetrieveObjectGridCac…** and in the **Properties** view below the canvas, select the **Details** tab on the left.



Checks WebSphere Object Grid for data : RetrieveObjectGridCache1

Catalog Endpoint: cannot be empty.

Catalog Endpoint:*
Grid name:*
Map name:*
user:*
password:*
clientProperties:*
DataObject key to retrieve from the cache:* /body
XPath to DataObject to retrieve from the cache:* /body

____ 39. Fill out the fields in the Detail tab:

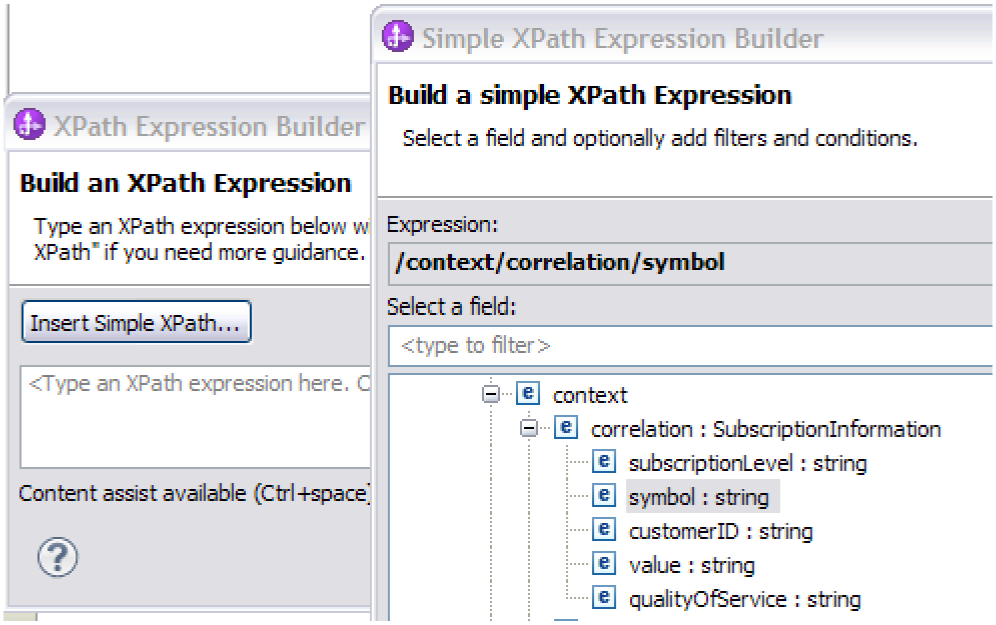
__ a. For Catalog Endpoint, provide the `hostname:port` of the XC10 appliance you are connecting to. Typically just the IP address is sufficient if the port used is the default of 2809.

__ b. Provide the XC10 grid name and map name to be used. The grid must already be created. For the map name, append ".LAT" to the grid name; this will allow data in cache to default to a 1 hour time-to-live. Do not append ".LAT" to the grid name if you want to remove the data in cache yourself.

__ c. Provide the user ID and password to be used to access the data grid.

__ d. For the clientProperties field, provide a local file name that contains the client security properties. For this lab, make the properties file empty.

__ e. For the DataObject key field, use the **Custom XPath** button, remove “/body” from the path, and then select the **Insert Simple XPath** button. Navigate to the **symbol** field under c**ontext** and **correlation**, select it, select OK, and OK again.



XPath Expression Builder

Build an XPath Expression

Type an XPath expression below w XPath" if you need more guidance.

<Type an XPath expression here. C

Content assist available (Ctrl+space)

Simple XPath Expression Builder

Build a simple XPath Expression

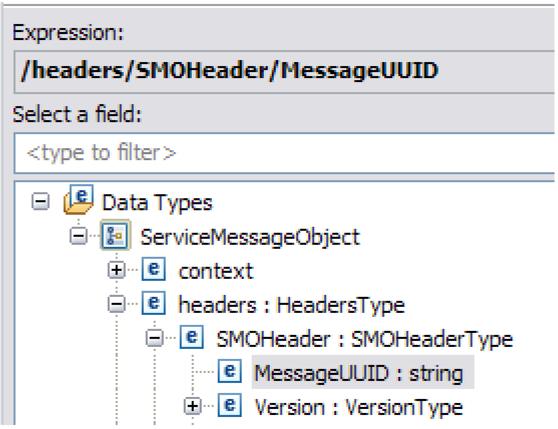
Select a field and optionally add filters and conditions.

Expression: /context/correlation/symbol

Select a field: <type to filter>

Insert Simple XPath...
context
correlation : SubscriptionInformation
subscriptionLevel : string
symbol : string
customerID : string
value : string
qualityOfService : string

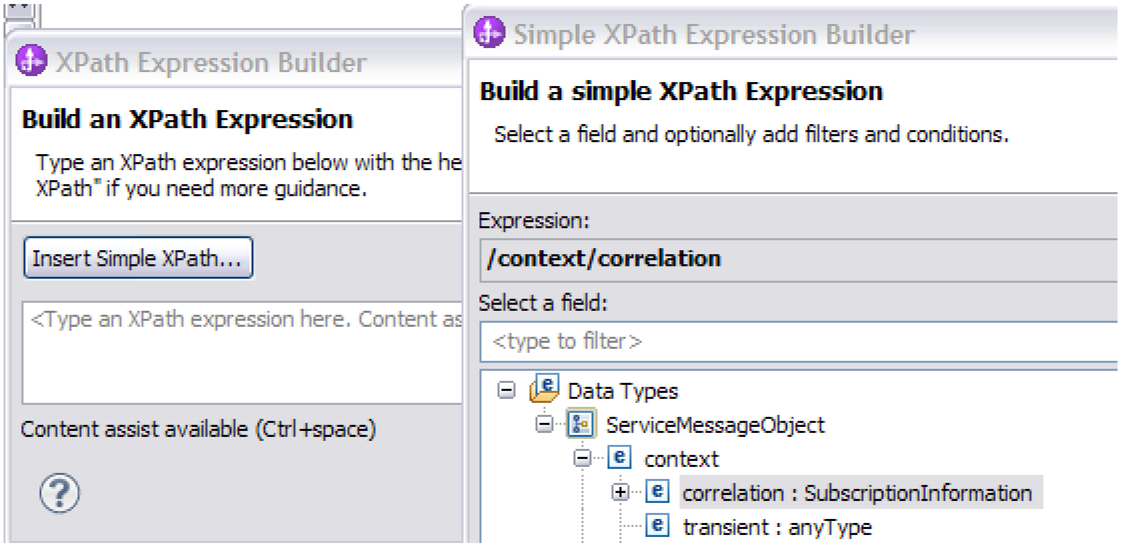
NOTE: if you are using the **MessageUUID** as the key, then expand the path to the field as below:



Expression: /headers/SMOHeader/MessageUUID

Select a field: <type to filter>

__ f. For the XPath to the DataObject field, use the **Custom XPath** button, remove “/body” from the path, and then select the **Insert Simple XPath** button. Navigate to **correlation: SubscriptionInformation** under c**ontext**, select it, select OK, and OK again.



XPath Expression Builder

Build an XPath Expression

Type an XPath expression below with the he
XPath" if you need more guidance.

<Type an XPath expression here. Content as

Content assist available (Ctrl+space)

Simple XPath Expression Builder

Build a simple XPath Expression

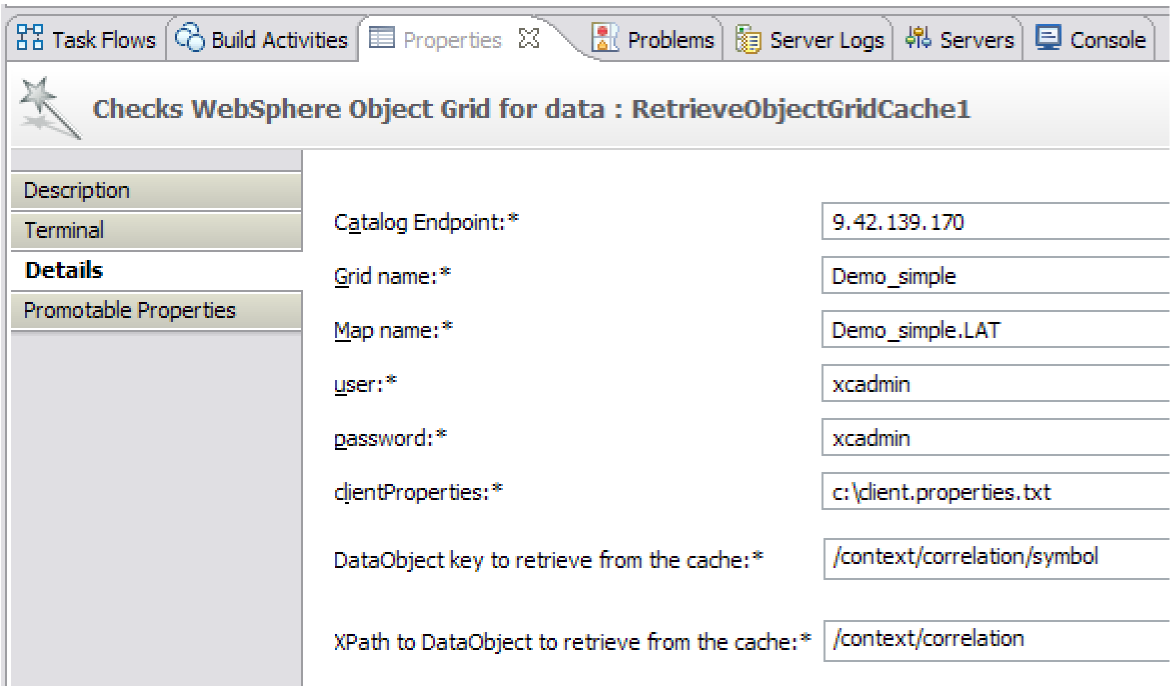
Select a field and optionally add filters and conditions.

Expression: **/context/correlation**

Select a field: <type to filter>

```
⊟ Data Types
   ⊟ ServiceMessageObject
      ⊟ context
         ⊞ correlation : SubscriptionInformation
           transient : anyType
```

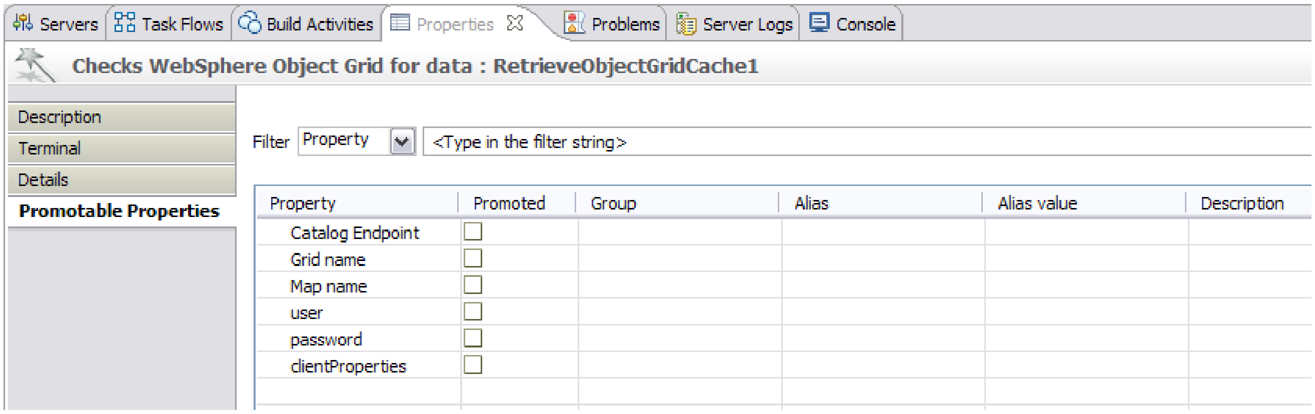
__ g. You should now have the following for the Properties Details for **RetrieveObjectGridCache1**.



Checks WebSphere Object Grid for data : RetrieveObjectGridCache1

Catalog Endpoint:*	9.42.139.170
Grid name:*	Demo_simple
Map name:*	Demo_simple.LAT
user:*	xcadmin
password:*	xcadmin
clientProperties:*	c:\client.properties.txt
DataObject key to retrieve from the cache:*	/context/correlation/symbol
XPath to DataObject to retrieve from the cache:*	/context/correlation

____ 40. (Optional) If you want to be able to configure these values with the server administration console after the application is installed, go to the **Promotable Properties** tab, and select the properties to be promoted.



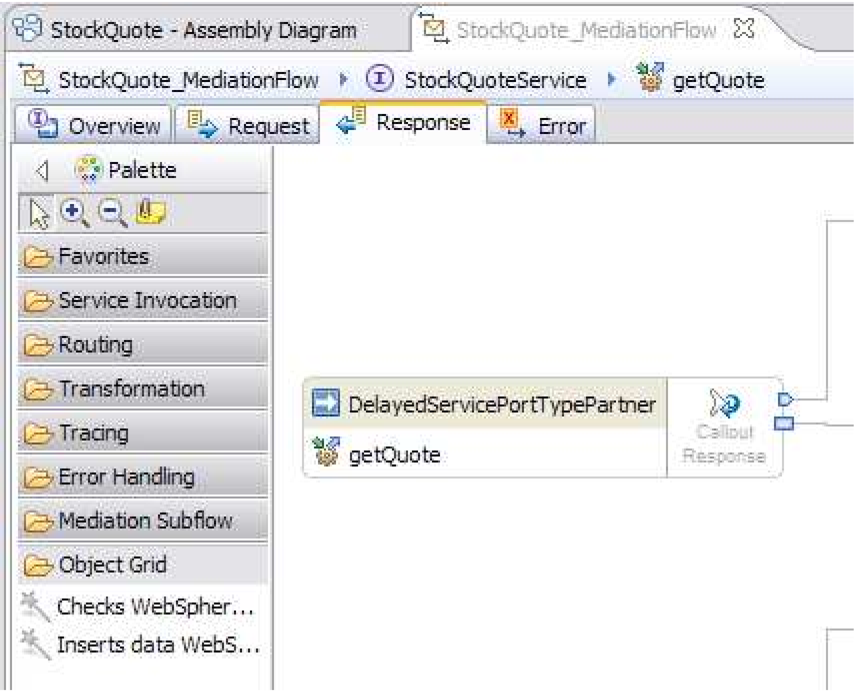
Checks WebSphere Object Grid for data : RetrieveObjectGridCache1

Filter Property <Type in the filter string>

Property	Promoted	Group	Alias	Alias value	Description
Catalog Endpoint	□				
Grid name	□				
Map name	□				
user	□				
password	□				
clientProperties	□				

____ 41. Save your changes using **File** and **Save All**.

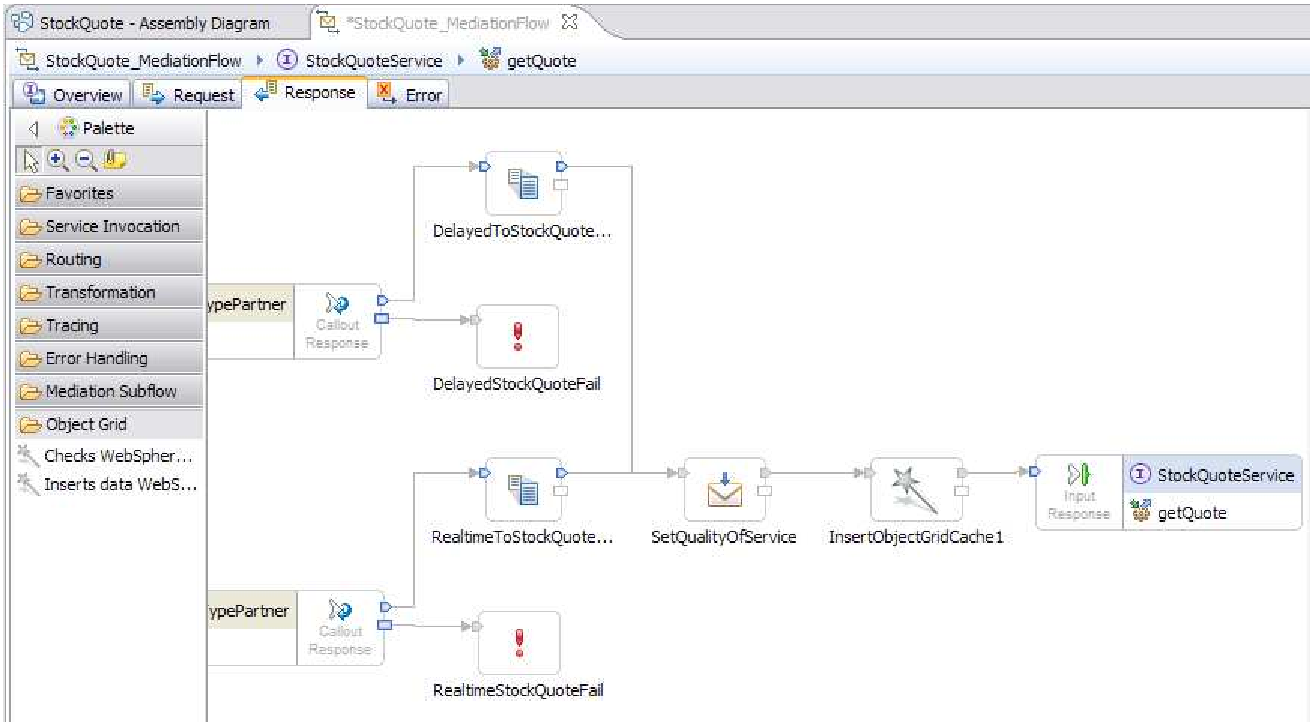
____ 42. Now you need to update the **Response** part of this mediation flow. Click the **Response** tab to display the **Response** flow on the canvas.



____ 43. On the **Response** flow, drag the **Inserts data … for caching** primitive onto the **Response** flow canvas. Right click the connection between **SetQualityOfService** and **getQuote: StockQuoteService** and delete it. Wire the **out** terminal of **SetQualityOfService** to the **in** terminal of **InsertObjectCache1**. Wire the **out** terminal of **InsertObjectCache1** to the **in** terminal of **getQuote: StockQuoteService**. Right click the canvas, and ensure **Automatic Layout** is selected.

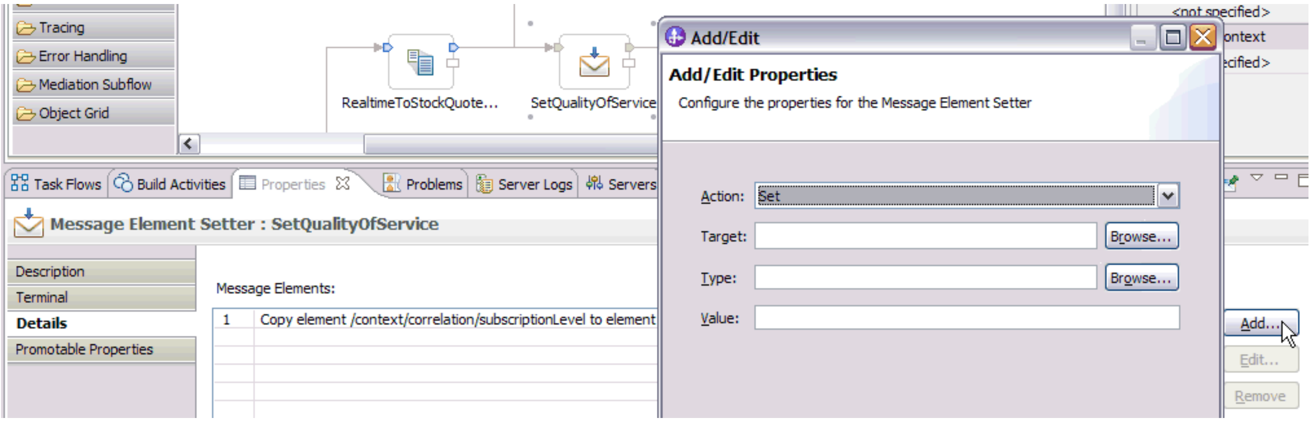
NOTE: that at the time of writing with the available version of com.ibm.bpm.OGCache_1.0.0.jar, the **fail** terminal is not used at all in **InsertObjectGridCac…**

NOTE: that this **InsertObjectGridCac…** primitive will replace the data in the grid if an object with the key value already exists, without any feedback to you if it does so. If you do not want to use objects that are inserted into the grid by another mediation flow, then as discussed earlier, you may want to look at using the field **MessageUUID** as the key to the data grid objects to isolate the object to the use of this mediation only.



Similar to the **Request** flow, you now need a **Message Element Setter** primitive to copy data into the **SubscriptionInformation** business object from the **body** into the **context correlation** so as to be available to both the **Request** and **Response** flows for the **ObjectGrid** primitives. In this sample, on the **Response** side the existing **SetQualityofService** primitive is a **Message Element Setter** primitive, so you will modify it to copy the 2 additional fields (**value** and **qualityOfService**) from the body into the context correlation.

____ 44. Double-click the **SetQualityofService** primitive, and go to the **Properties** view below the canvas. In the **Properties** view, click **Details**. On the right, click **Add** to **Add/Edit Properties**.



Message Element Setter : SetQualityOfService

Message Elements:

1 Copy element /context/correlation/subscriptionLevel to element

Add/Edit

Add/Edit Properties

Configure the properties for the Message Element Setter

Action: Set

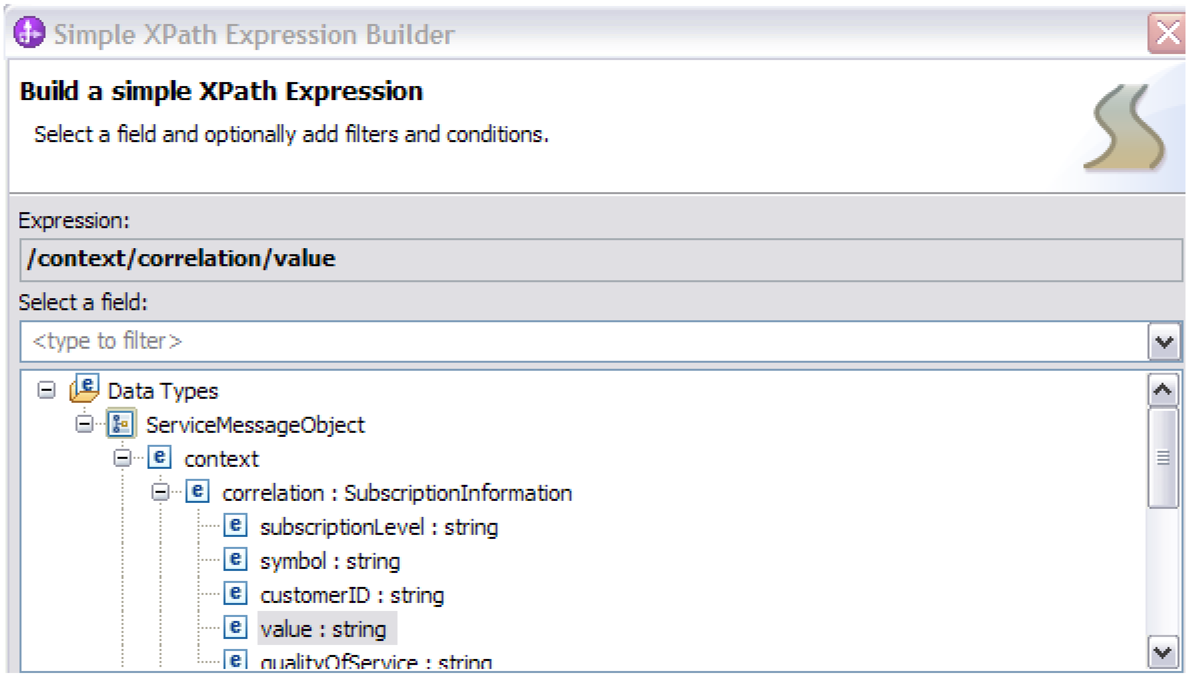
Target:

Type:

Value:

____ 45. On the **Action** pulldown, select **Copy**. You will copy the two fields - **value** and **qualityOfService** - from the **body** of the **StockQuoteRequest** business object to the **context correlation** of the **SubscriptionInformation**.

__ a. For the **Target**, click **Browse** to build the XPath expression. Expand the **context** part of the path to get to **value**, and select OK.



Simple XPath Expression Builder

Build a simple XPath Expression

Select a field and optionally add filters and conditions.

Expression:

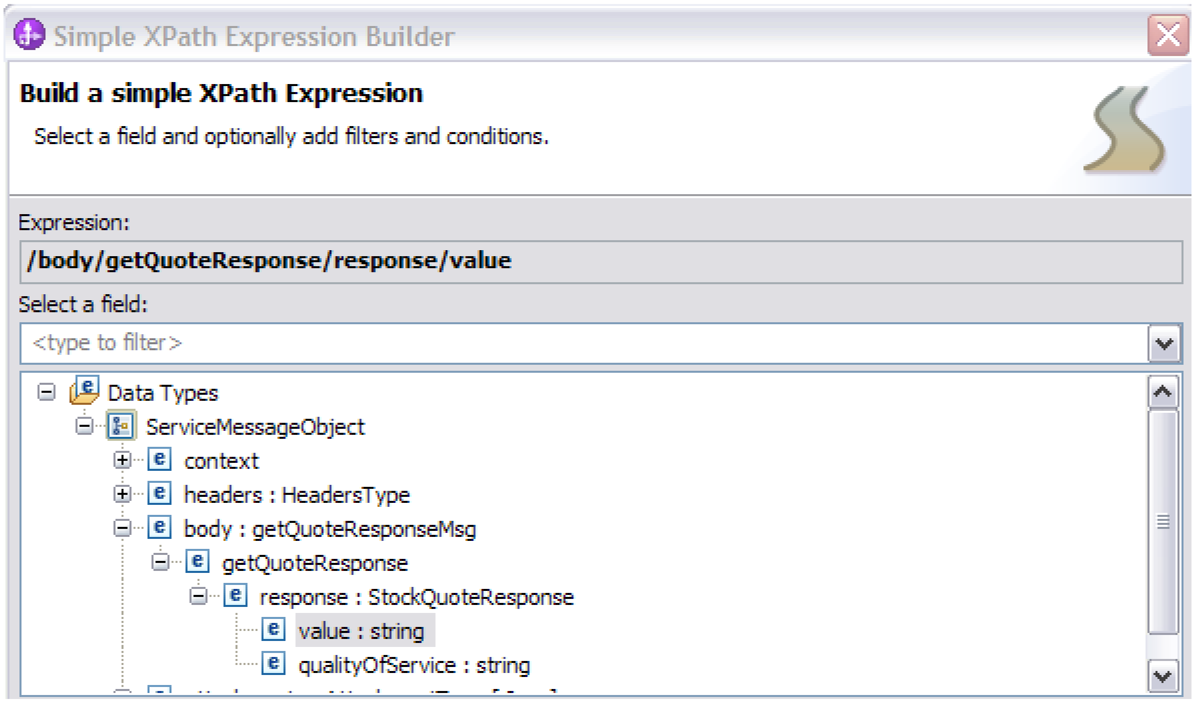
/context/correlation/value

Select a field:

<type to filter>

- Data Types
  - ServiceMessageObject
    - context
      - correlation : SubscriptionInformation
        - subscriptionLevel : string
        - symbol : string
        - customerID : string
        - value : string
        - qualityOfService : string

__ b. Similarly, for the **Source**, click **Browse**, expand the **body** portion of the path, select **value**, and OK and then **Finish**.



Simple XPath Expression Builder

Build a simple XPath Expression

Select a field and optionally add filters and conditions.

Expression:

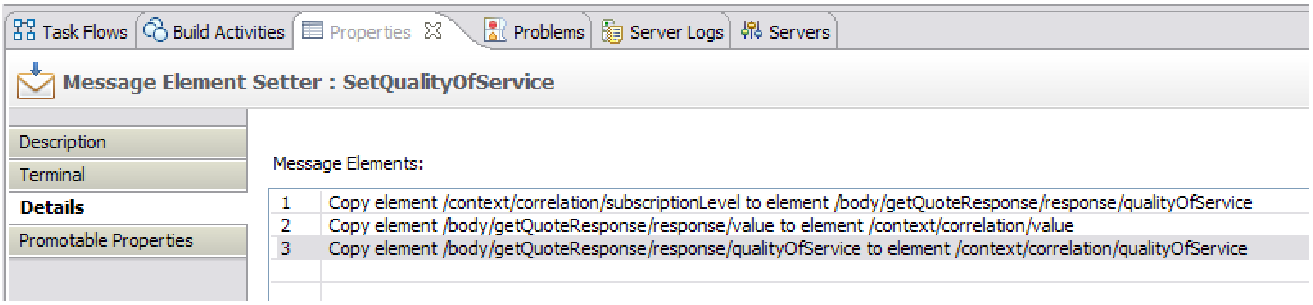
/body/getQuoteResponse/response/value

Select a field:

<type to filter>

- Data Types
  - ServiceMessageObject
    - context
    - headers : HeadersType
    - body : getQuoteResponseMsg
      - getQuoteResponse
        - response : StockQuoteResponse
          - value : string
          - qualityOfService : string

__ c. Repeat the steps to **Add** another **copy** element for the **qualityOfService** field. You should now have the following **Detail** properties for **SetQualityofService.**

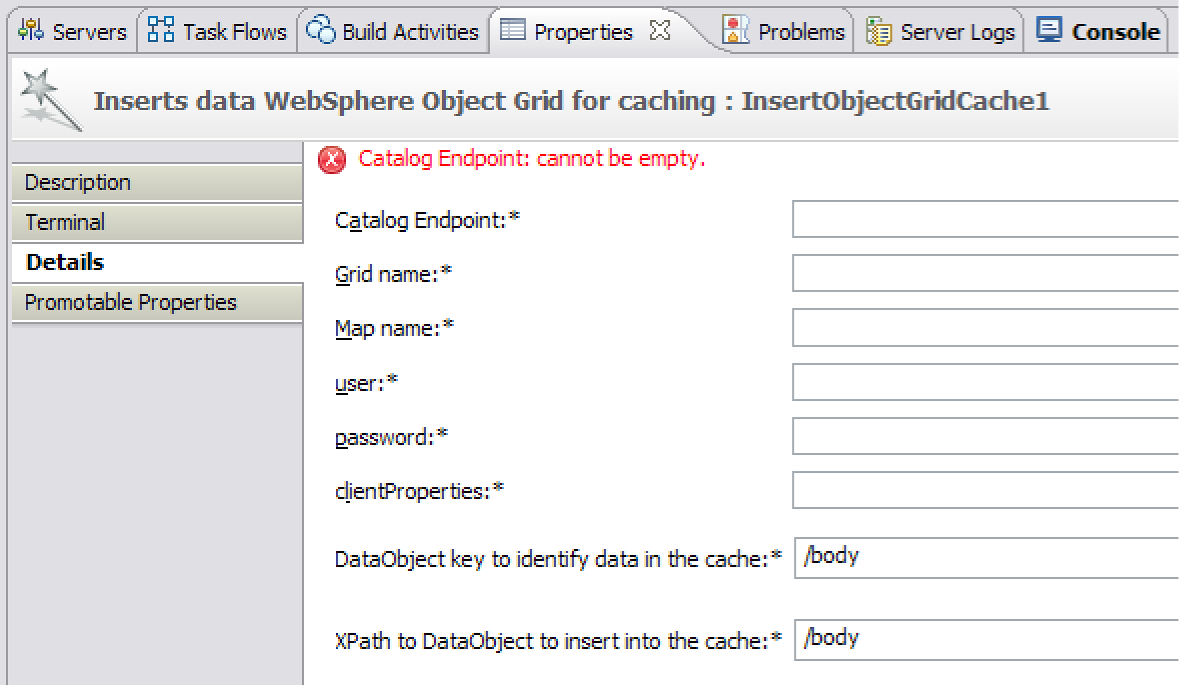


Message Element Setter : SetQualityOfService

Message Elements:

1	Copy element /context/correlation/subscriptionLevel to element /body/getQuoteResponse/response/qualityOfService
2	Copy element /body/getQuoteResponse/response/value to element /context/correlation/value
3	Copy element /body/getQuoteResponse/response/qualityOfService to element /context/correlation/qualityOfService

____ 46. Double-click **InsertObjectGridCac…** and in the **Properties** view below the canvas, select the **Details** tab.

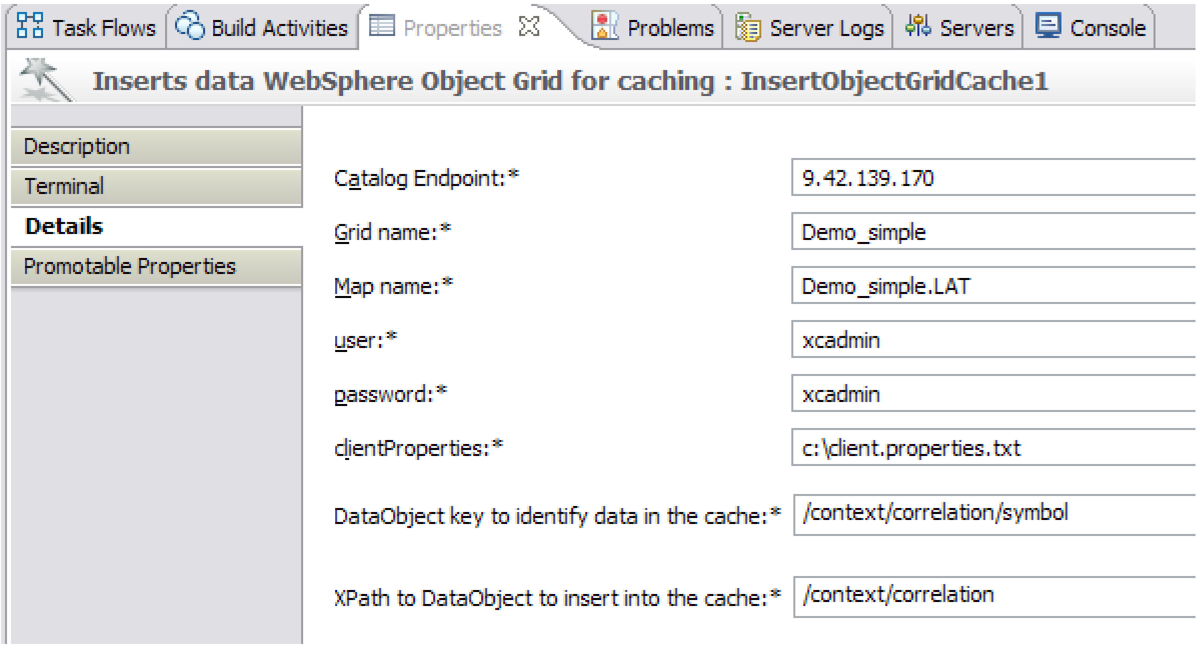


Inserts data WebSphere Object Grid for caching : InsertObjectGridCache1

Catalog Endpoint: cannot be empty.

Catalog Endpoint:*	
Grid name:*	
Map name:*	
user:*	
password:*	
clientProperties:*	
DataObject key to identify data in the cache:*	/body
XPath to DataObject to insert into the cache:*	/body

__ a. Fill out the fields in the Detail tab exactly as you did in step 39 above (____ 39). You should now see the **Properties Details** for **InsertObjectGridCache1** as below.



Inserts data WebSphere Object Grid for caching : InsertObjectGridCache1

Catalog Endpoint:* 9.42.139.170

Grid name:* Demo_simple

Map name:* Demo_simple.LAT

user:* xcadmin

password:* xcadmin

clientProperties:* c:\client.properties.txt

DataObject key to identify data in the cache:* /context/correlation/symbol

XPath to DataObject to insert into the cache:* /context/correlation

____ 47. (Optional) As in step 40 above (____ 40), if you want to be able to configure these values with the server administration console after the application is installed, go to the **Promotable Properties** tab, and select the properties to be promoted.

____ 48. Save your changes using **File** and **Save All**. Close all the editors you have open.

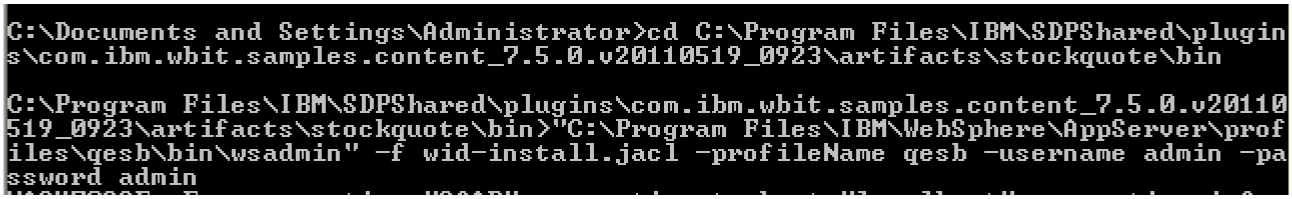
____ 49. Restart the WebSphere Enterprise Service Bus server.

Now you are ready to install the runtime components. Detailed instructions for this runtime install are provided in the information center that is presented to you when you go to the **View Instructions** mentioned in the introduction of this section. A summary of the steps are provided below; if you have difficulties with these steps, see the details in the information center for this Stock Quote sample.

____ 50. Ensure the WebSphere Enterprise Service Bus server is started - see step 23 above (____ 23) if necessary.

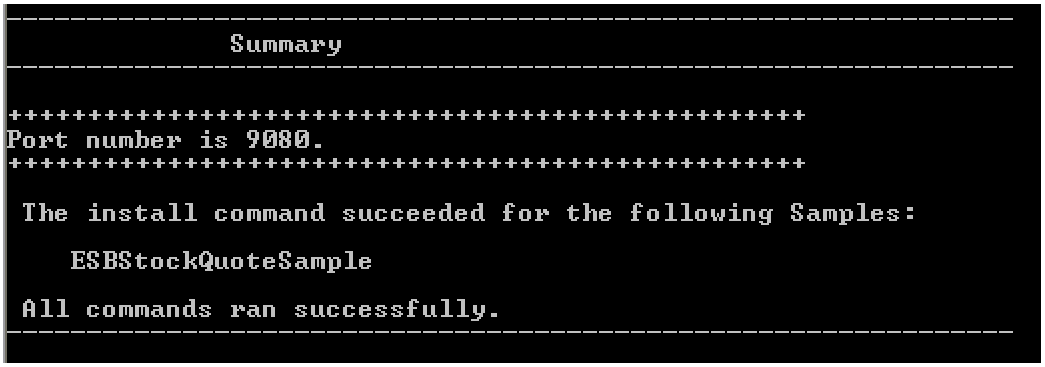
____ 51. Open a command prompt and switch to the SDPShared directory to …plugins/com.ibm.wbit.samples.content/artifacts/stockquote/bin/. If there are multiple versions of com.ibm.wbit.samples.content, then choose the latest version. In this lab’s environment, the SDPShared directory is: C:\Program Files\IBM\SDPShared.

____ 52. While in this directory, execute the following wsadmin command: “wsadmin -f wid-install.jacl -profileName ***esb_profile_name*** -username ***username*** -password ***password***”. See below for how this was run for this lab. This step only needs to be run once, regardless of further mediation changes made to the Stock Quote sample.



```
C:\Documents and Settings\Administrator>cd C:\Program Files\IBM\SDPShared\plugin
s\com.ibm.wbit.samples.content_7.5.0.v20110519_0923\artifacts\stockquote\bin

C:\Program Files\IBM\SDPShared\plugins\com.ibm.wbit.samples.content_7.5.0.v20110
519_0923\artifacts\stockquote\bin>"C:\Program Files\IBM\WebSphere\AppServer\prof
iles\qesb\bin\wsadmin" -f wid-install.jacl -profileName qesb -username admin -pa
ssword admin
```

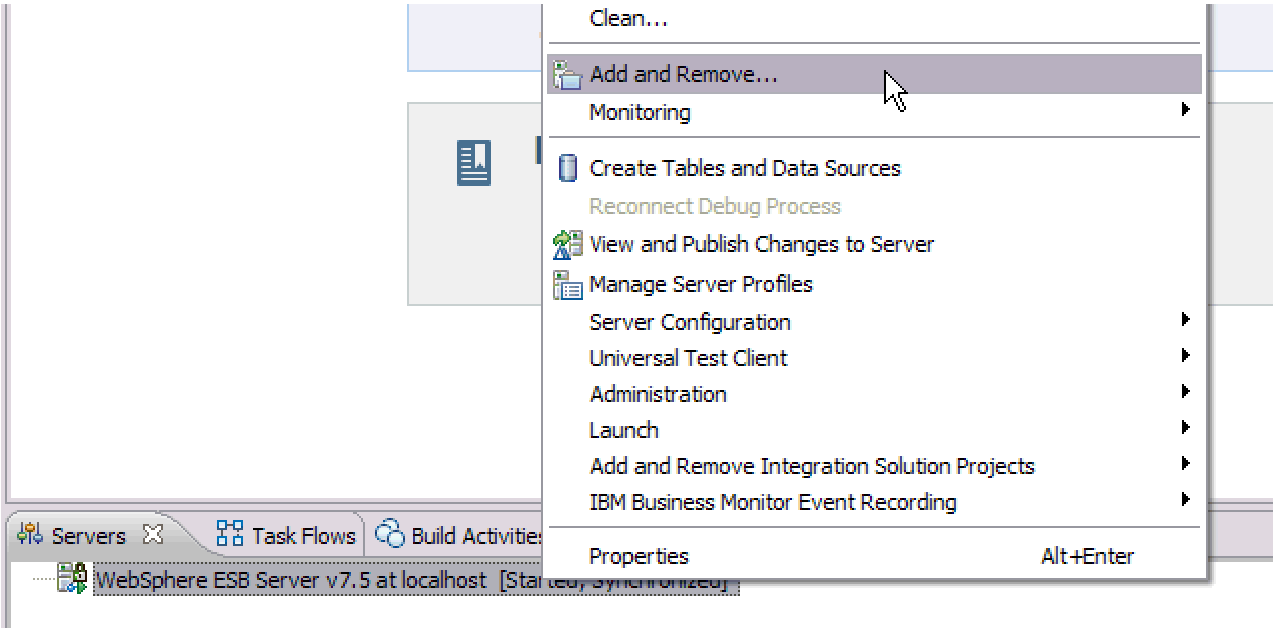
____ 53. The command results should end with the following summary. If you do not see this summary or a port number of 9080, then see the information center sample runtime documentation for more details.



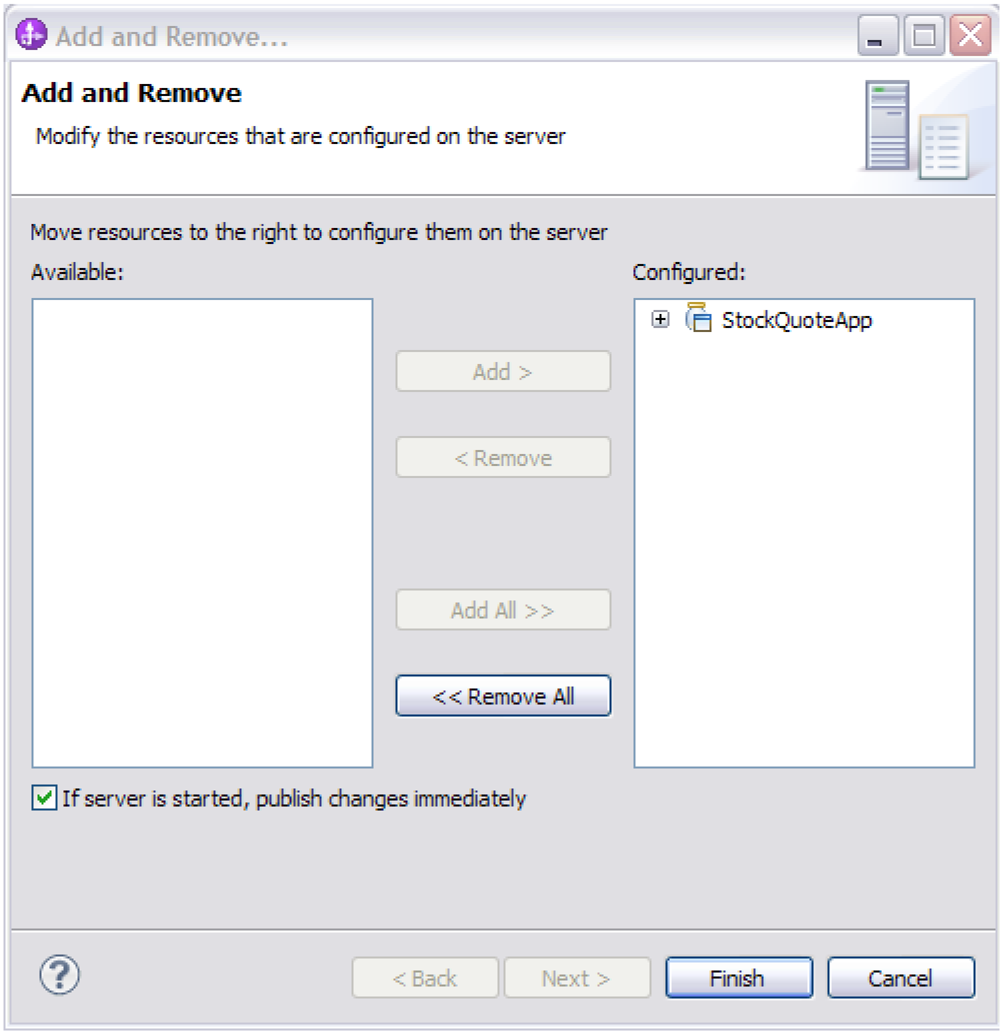
```
---------------------------------------------------------------
              Summary
---------------------------------------------------------------

++++++++++++++++++++++++++++++++++++++++++++++++++
Port number is 9080.
++++++++++++++++++++++++++++++++++++++++++++++++++

 The install command succeeded for the following Samples:

    ESBStockQuoteSample

 All commands ran successfully.
---------------------------------------------------------------
```

____ 54. Add the StockQuoteApp to the running WebSphere Enterprise Service Bus server:

__ a. Right-click the server and select **Add and Remove**.


Clean...
Add and Remove...
Monitoring
Create Tables and Data Sources
Reconnect Debug Process
View and Publish Changes to Server
Manage Server Profiles
Server Configuration
Universal Test Client
Administration
Launch
Add and Remove Integration Solution Projects
IBM Business Monitor Event Recording
Properties
Alt+Enter
Servers
Task Flows
Build Activities
WebSphere ESB Server v7.5 at localhost [Started, Synchronized]

__ b. On the resulting **Add and Remove** popup window, click **Add All**, and then click on **Finish**.



Add and Remove

Modify the resources that are configured on the server

Move resources to the right to configure them on the server

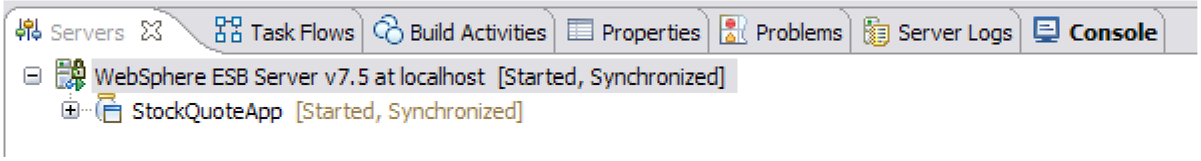
Available:

Configured:

⊞ StockQuoteApp

☑ If server is started, publish changes immediately

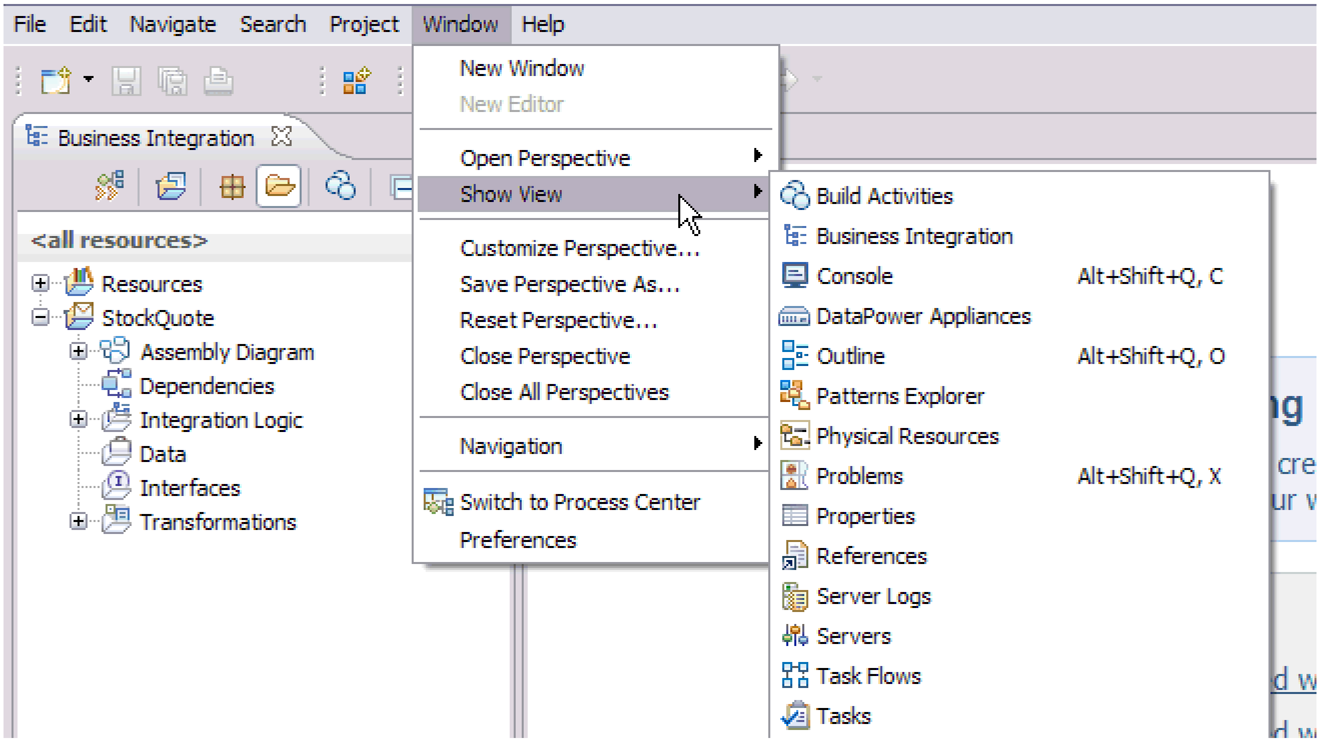
__ c. Wait for the publishing to complete, which is when both the server and application are in a "Started, Synchronized" state.

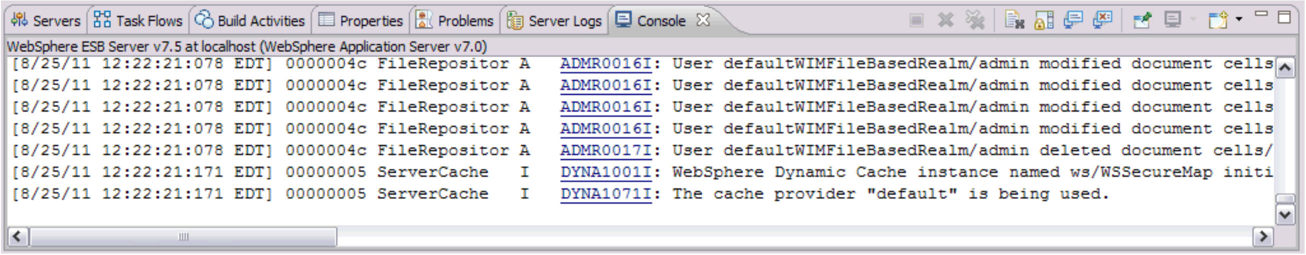


⊟ WebSphere ESB Server v7.5 at localhost [Started, Synchronized]

⊞ StockQuoteApp [Started, Synchronized]

__ d. (Optional) If you do not have some of the views as above, such as Console or Server Logs, go to **Window**, select **Show View**, and find the views of your choice. Note the **Console** view is especially useful to see the server console messages.



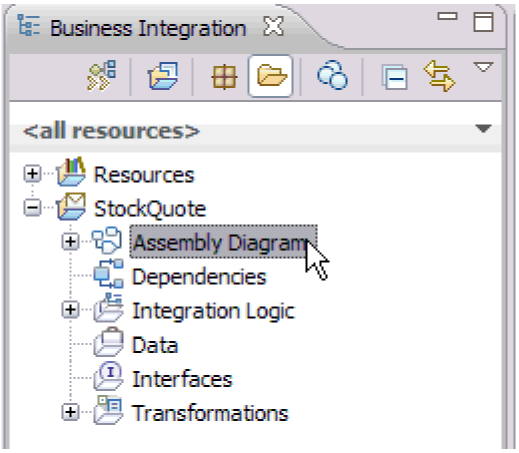


WebSphere ESB Server v7.5 at localhost (WebSphere Application Server v7.0)

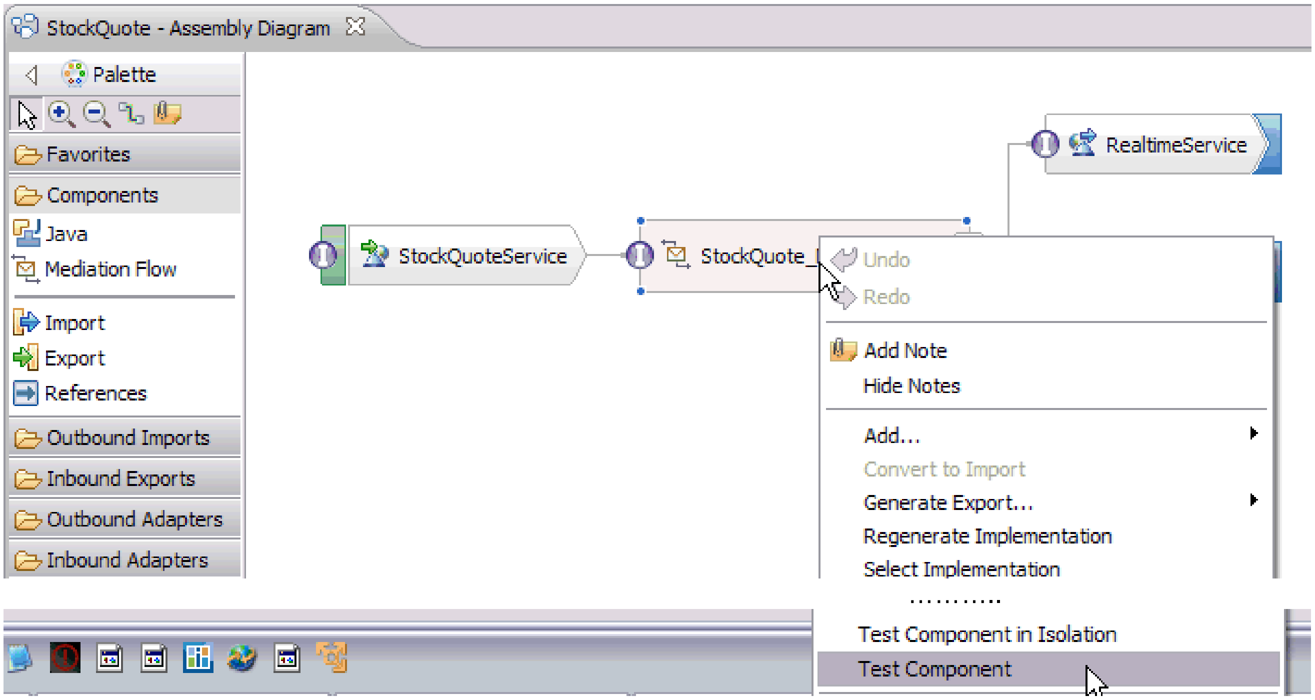
```
[8/25/11 12:22:21:078 EDT] 0000004c FileRepositor A   ADMR0016I: User defaultWIMFileBasedRealm/admin modified document cells
[8/25/11 12:22:21:078 EDT] 0000004c FileRepositor A   ADMR0016I: User defaultWIMFileBasedRealm/admin modified document cells
[8/25/11 12:22:21:078 EDT] 0000004c FileRepositor A   ADMR0016I: User defaultWIMFileBasedRealm/admin modified document cells
[8/25/11 12:22:21:078 EDT] 0000004c FileRepositor A   ADMR0016I: User defaultWIMFileBasedRealm/admin modified document cells
[8/25/11 12:22:21:078 EDT] 0000004c FileRepositor A   ADMR0017I: User defaultWIMFileBasedRealm/admin deleted document cells/
[8/25/11 12:22:21:171 EDT] 00000005 ServerCache   I   DYNA1001I: WebSphere Dynamic Cache instance named ws/WSSecureMap initi
[8/25/11 12:22:21:171 EDT] 00000005 ServerCache   I   DYNA1071I: The cache provider "default" is being used.
```

Now you are ready to actually test this modified Stock Quote sample. This lab assumes you have a simple data grid called Demo_simple that is empty.

____ 55. In the **Business Integration** pane on the left side, double-click **Assembly Diagram** under the **StockQuote** mediation module.

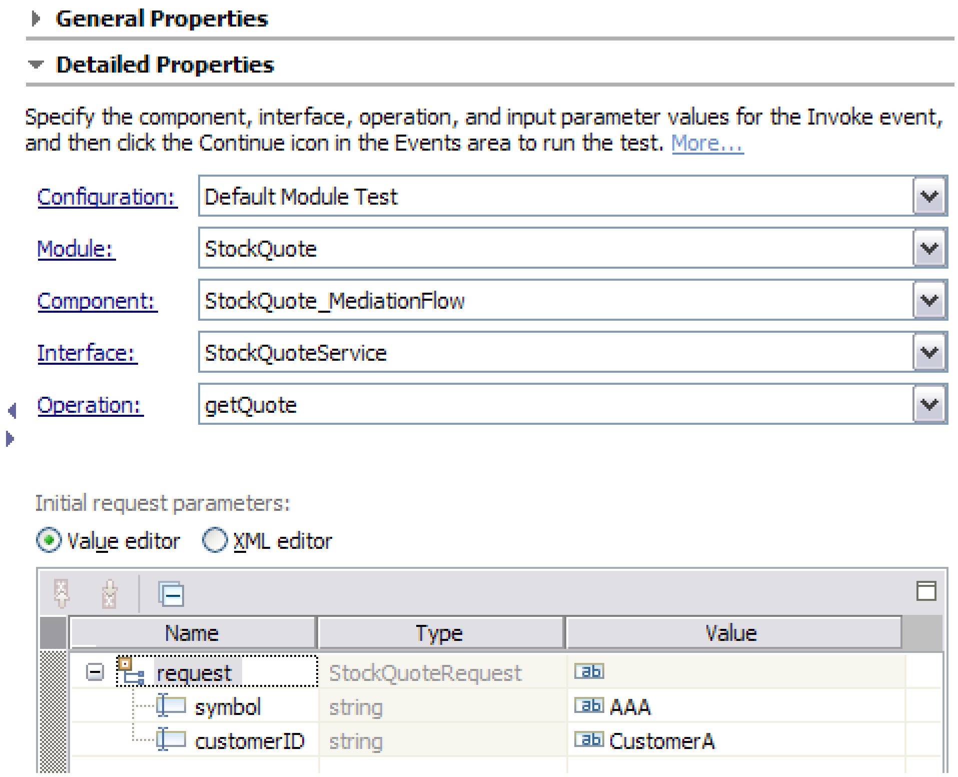


____ 56. Right-click the **StockQuote_MediationFlow** component in the editor, and select **Test Component**.



____ 57. You should now see a **StockQuote_Test** canvas. On the right side of the canvas, under the **Detailed Properties**, you will see the **Initial request parameters**. Here, enter information by double clicking the cell in the **Value** column, for both the **symbol** and **customerID** rows. For the symbol enter **AAA**. For the customerID enter **CustomerA**.

NOTE that this sample comes with a database pre-loaded with some customer information. Therefore do not deviate from the values provided in this lab or in the instructions in the information center document for this sample.



General Properties

Detailed Properties

Specify the component, interface, operation, and input parameter values for the Invoke event, and then click the Continue icon in the Events area to run the test. More...

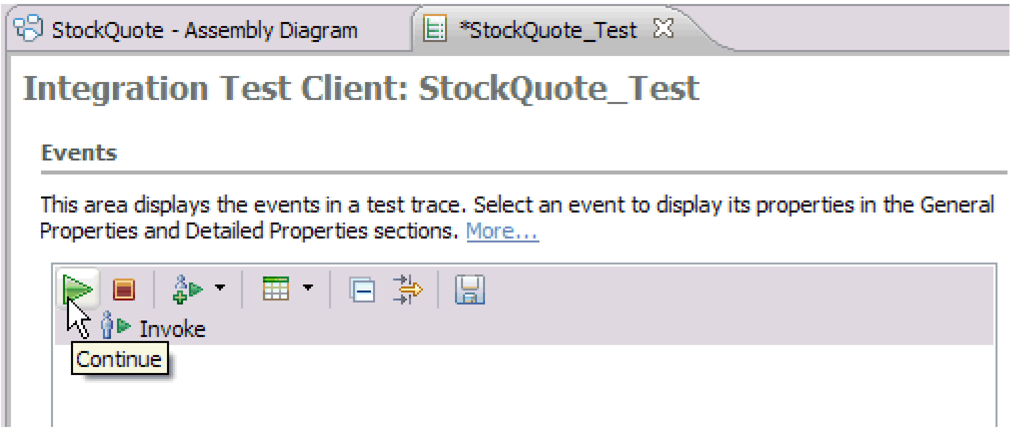
Configuration:	Default Module Test
Module:	StockQuote
Component:	StockQuote_MediationFlow
Interface:	StockQuoteService
Operation:	getQuote

Initial request parameters:

● Value editor ○ XML editor

Name	Type	Value
request	StockQuoteRequest	
symbol	string	AAA
customerID	string	CustomerA

____ 58. On the left side of the **StockQuote_Test** canvas, click the **Continue** green right arrow to invoke getQuote.

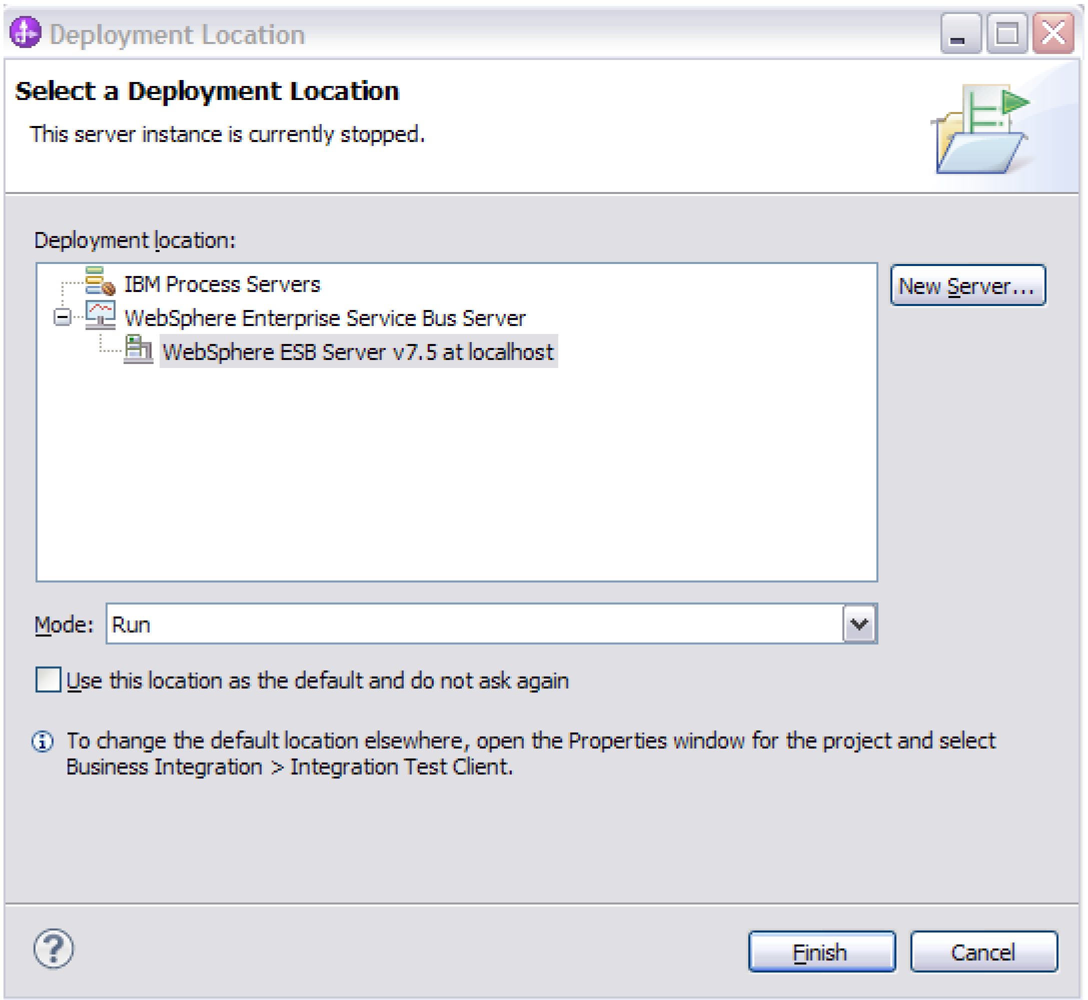


Integration Test Client: StockQuote_Test

Events

This area displays the events in a test trace. Select an event to display its properties in the General Properties and Detailed Properties sections. More...

____ 59. A **Deployment Location** popup window will appear. Select the **WebSphere ESB server v7.5 at localhost**. If you do not want to see this screen again, also check the Use **this location as the default …** box. Select **Finish**.



Deployment Location

Select a Deployment Location

This server instance is currently stopped.

Deployment location:

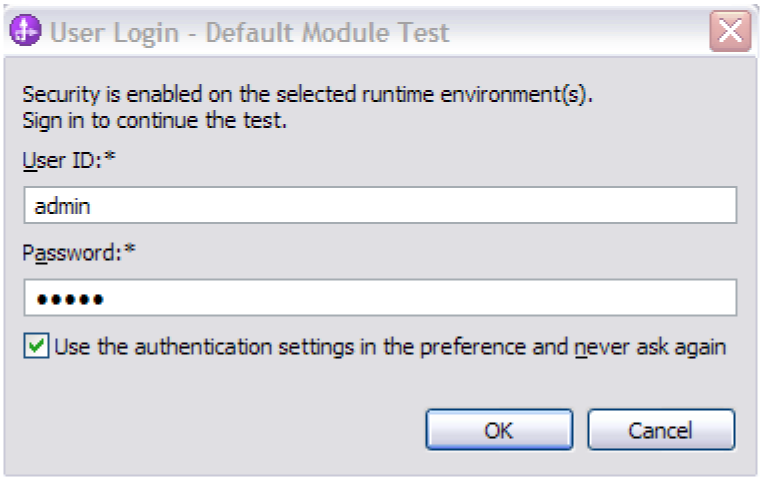
- IBM Process Servers
- WebSphere Enterprise Service Bus Server
  - WebSphere ESB Server v7.5 at localhost

Mode: Run

□ Use this location as the default and do not ask again

To change the default location elsewhere, open the Properties window for the project and select Business Integration > Integration Test Client.

____ 60. A **User Logon** popup window will appear. Enter **admin** for both the ID and password. If you never want to fill this out again, check the Use **the authentication settings….** box, and then select **OK**.



User Login - Default Module Test

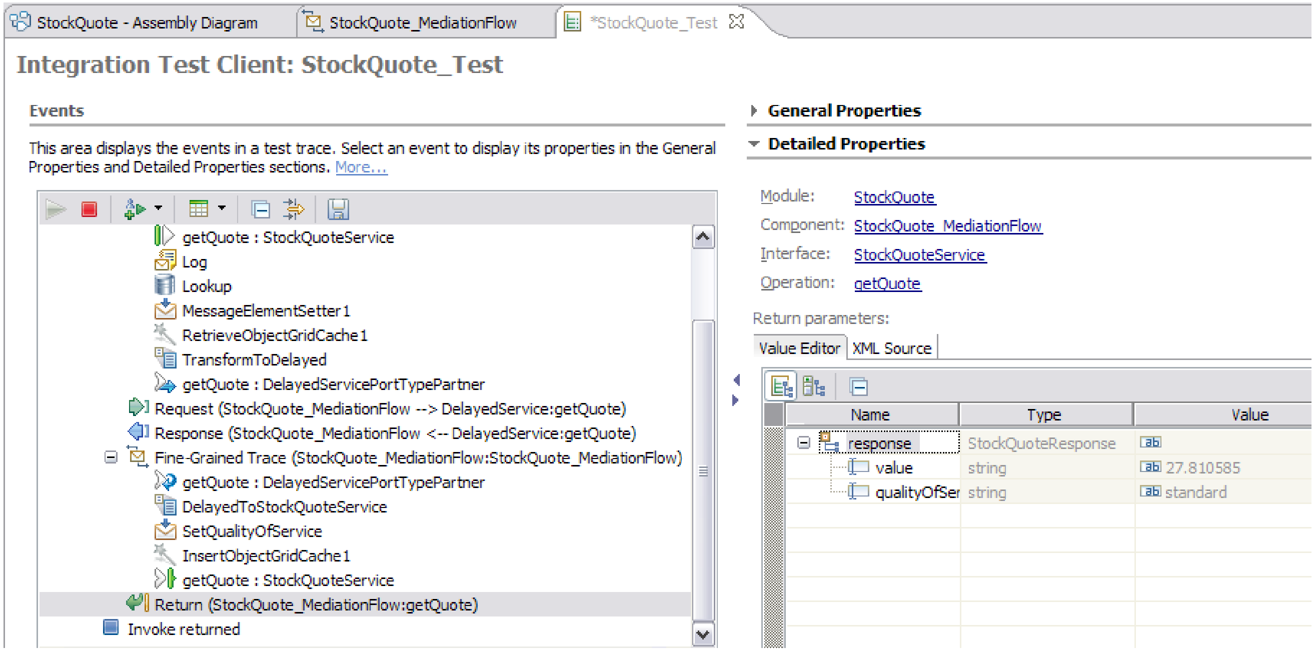
Security is enabled on the selected runtime environment(s).
Sign in to continue the test.

User ID:* admin

Password:* •••••

☑ Use the authentication settings in the preference and never ask again

____ 61. Under some conditions, the server may be restarted and/or the application may be republished, so be patient and watch the **Servers** view. The test results should look similar to below. Note that the flow took the **Delayed** path because the data attempted to be retrieved from the grid did not exist.


StockQuote - Assembly Diagram | StockQuote_MediationFlow | *StockQuote_Test

Integration Test Client: StockQuote_Test

Events

This area displays the events in a test trace. Select an event to display its properties in the General Properties and Detailed Properties sections. More...

```
getQuote : StockQuoteService
Log
Lookup
MessageElementSetter1
RetrieveObjectGridCache1
TransformToDelayed
getQuote : DelayedServicePortTypePartner
Request (StockQuote_MediationFlow --> DelayedService:getQuote)
Response (StockQuote_MediationFlow <-- DelayedService:getQuote)
Fine-Grained Trace (StockQuote_MediationFlow:StockQuote_MediationFlow)
    getQuote : DelayedServicePortTypePartner
    DelayedToStockQuoteService
    SetQualityOfService
    InsertObjectGridCache1
    getQuote : StockQuoteService
Return (StockQuote_MediationFlow:getQuote)
Invoke returned
```

General Properties

Detailed Properties

Module: StockQuote
Component: StockQuote_MediationFlow
Interface: StockQuoteService
Operation: getQuote

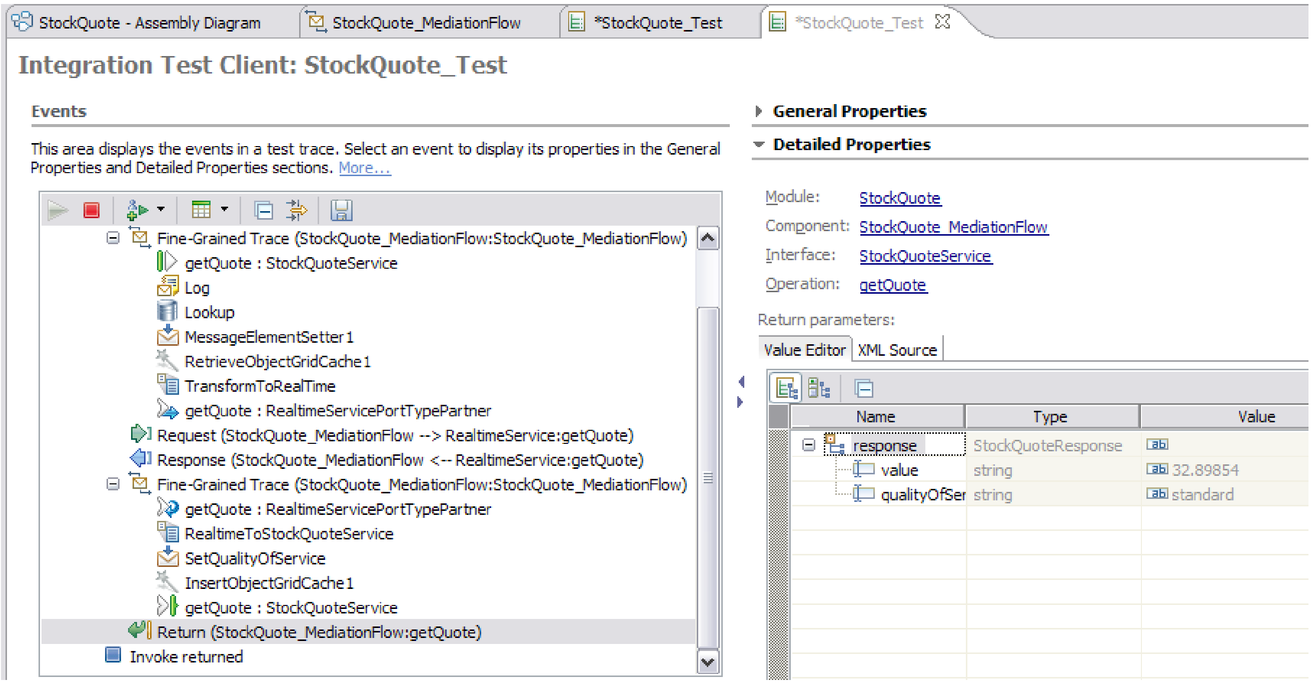
Return parameters:

Value Editor | XML Source

Name	Type	Value
response	StockQuoteResponse	
value	string	27.810585
qualityOfSer	string	standard

____ 62. Do exactly the same test again. This time note that the flow took the **Real-time** path, because the data attempted to be retrieved from the grid now does exist, as it was inserted on the first test.

NOTE: For simplicity of demonstration purposes, the data in the cache was not actually used in the flow, and therefore the field **value** has a different value below on the second test. It is left to you as an exercise to further modify this mediation flow to actually use the data retrieved from cache and to remove the **Real-time** web service call from the mediation flow when the data is found in cache.


StockQuote - Assembly Diagram | StockQuote_MediationFlow | *StockQuote_Test | *StockQuote_Test

Integration Test Client: StockQuote_Test

Events

This area displays the events in a test trace. Select an event to display its properties in the General Properties and Detailed Properties sections. More...

```
Fine-Grained Trace (StockQuote_MediationFlow:StockQuote_MediationFlow)
    getQuote : StockQuoteService
    Log
    Lookup
    MessageElementSetter1
    RetrieveObjectGridCache1
    TransformToRealTime
    getQuote : RealtimeServicePortTypePartner
Request (StockQuote_MediationFlow --> RealtimeService:getQuote)
Response (StockQuote_MediationFlow <-- RealtimeService:getQuote)
Fine-Grained Trace (StockQuote_MediationFlow:StockQuote_MediationFlow)
    getQuote : RealtimeServicePortTypePartner
    RealtimeToStockQuoteService
    SetQualityOfService
    InsertObjectGridCache1
    getQuote : StockQuoteService
Return (StockQuote_MediationFlow:getQuote)
Invoke returned
```

General Properties

Detailed Properties

Module: StockQuote
Component: StockQuote_MediationFlow
Interface: StockQuoteService
Operation: getQuote

Return parameters:

Value Editor | XML Source

Name	Type	Value
response	StockQuoteResponse	
value	string	32.89854
qualityOfSer	string	standard

____ 63. Observe the various **monitor** results for your simple data grid with the XC10 web console.

Appendix A: The SimpleGridDemo Java application

This Java application is a variation of the sample provided here: **http://publib.boulder.ibm.com/infocenter/wdpxc/v2r0/topic/com.ibm.websphere.datapower.xc.doc/rsimplecache.html.**

```
//Section 1 start *********************************************************
import com.ibm.websphere.objectgrid.ClientClusterContext;
import com.ibm.websphere.objectgrid.CopyMode;
import com.ibm.websphere.objectgrid.ObjectGrid;
import com.ibm.websphere.objectgrid.ObjectGridException;
import com.ibm.websphere.objectgrid.ObjectGridManager;
import com.ibm.websphere.objectgrid.ObjectGridManagerFactory;
import com.ibm.websphere.objectgrid.ObjectGridRuntimeException;
import com.ibm.websphere.objectgrid.ObjectMap;
import com.ibm.websphere.objectgrid.Session;
import com.ibm.websphere.objectgrid.plugins.TransactionCallbackException;
import com.ibm.websphere.objectgrid.security.config.ClientSecurityConfiguration;
import
com.ibm.websphere.objectgrid.security.config.ClientSecurityConfigurationFactory;
import com.ibm.websphere.objectgrid.security.plugins.CredentialGenerator;
import
com.ibm.websphere.objectgrid.security.plugins.builtins.UserPasswordCredentialGene
rator;
//Section 1 end *********************************************************

public class SimpleGridDemo
{
//Section 2 start *********************************************************
      static String username = "xcadmin";
      static String password = "xcadmin";
      static String gridName = "Demo_simple";
      static String mapName  = "Demo_simple";
//    static String mapName  = "Demo_simple.LAT.P";
      static String hostName = "9.42.139.170";

      static ObjectGrid clientGrid = null;
      static synchronized public ObjectGrid getObjectGrid()
      {
            if (clientGrid == null)
            {
                  ClientClusterContext ccc = null;
                  ObjectGridManager ogm = 
ObjectGridManagerFactory.getObjectGridManager();
                  ClientSecurityConfiguration clientSC = getAdminClientConfig();
                  try
                        {
                        ccc = ogm.connect(hostName+":2809", clientSC, null);
//                      ccc = ogm.connect(hostName+":2809", null, null);
                        }
                  catch (Throwable e)
                        {
                        System.out.println("--------------------------------------
------------------------");
                        System.out.println("Connection failure");
                        System.out.println("--------------------------------------
------------------------");
                        e.printStackTrace();
                        }
                  if (ccc != null)
                        {
                        try
                              {
                              clientGrid = ogm.getObjectGrid(ccc, gridName);
                              }
                        catch (ObjectGridRuntimeException ogre)
                              {
                              System.out.println("--------------------------------
------------------------------");
                              System.out.println("Object Grid failure");
                              System.out.println("--------------------------------
------------------------------");
                              ogre.printStackTrace();
                              }
                        }
            }
            return clientGrid;
      }

//NOTE *** that Word is hyphenating some words below – please remove hyphens when
running this code ***.
      public static ClientSecurityConfiguration getAdminClientConfig()
      {
            ClientSecurityConfiguration clientSC =
ClientSecurityConfigurationFactory.getClientSecurityConfiguration();
            clientSC.setSecurityEnabled(true);
            CredentialGenerator credGen = new
UserPasswordCredentialGenerator(username, password);
            clientSC.setCredentialGenerator(credGen);
            return clientSC;
      }
//Section 2 end *********************************************************

      public static void main(String args[]) throws Exception
      {
//Section 3 start *********************************************************
            System.out.println("------------------------------------------------
-------------");
            System.out.println("Simple Grid Test");
            System.out.println("------------------------------------------------
-------------");
            System.out.println("username     : "+username);
            System.out.println("password     : "+password);
            System.out.println("gridname     : "+gridName);
            System.out.println("mapname      : "+mapName);
            System.out.println("hostname     : "+hostName);
            System.out.println("------------------------------------------------
-------------");

            if (getObjectGrid() == null)
            {
                  System.out.println("-------------------------------------------
-------------------");
                  System.out.println("ERROR: unable to connect to objectgrid at
"+hostName);
                  System.out.println("-------------------------------------------
-------------------");
                  System.exit(1);
            }
            try
            {
            Session sess = clientGrid.getSession();
            ObjectMap map=sess.getMap(mapName);
//          map.setTimeToLive(60);
//          sess.begin();
            Object data = map.get("TestKey01");
            if (data!=null)
            {
                  System.out.println("-------------------------------------------
-------------------");
                  System.out.println("NOTE that the record with the following key
existed and was replaced: "+data);
                  System.out.println("-------------------------------------------
-------------------");
                  map.remove("TestKey01");
            }
            map.insert("TestKey01","TestValue01");
//          sess.commit();
            }
            catch (Exception e)
            {
                  System.out.println("-------------------------------------------
-------------------");
                  System.out.println("ERROR: failure working with objectgrid at
"+hostName);
                  System.out.println("-------------------------------------------
-------------------");
                  e.printStackTrace();
            }
//Section 3 end *********************************************************
      }
}
```

Appendix B: The JavaCompute node Java application

```
/*
 * Sample program for use with Product
 *  ProgIds: 5724-J06 5724-J05 5724-J04 5697-J09 5655-M74 5655-M75 5648-C63
 *  (C) Copyright IBM Corporation 2005.
 * All Rights Reserved * Licensed Materials - Property of IBM
 *
 * This sample program is provided AS IS and may be used, executed,
 * copied and modified without royalty payment by customer
 *
 * (a) for its own instruction and study,
 * (b) in order to develop applications designed to run with an IBM
 *     WebSphere product, either for customer's own internal use or for
 *     redistribution by customer, as part of such an application, in
 *     customer's own products.
 */

package com.ibm.broker.javacompute.samples;
import java.util.ListResourceBundle;
import java.util.regex.*;

import com.ibm.broker.javacompute.MbJavaComputeNode;
import com.ibm.broker.plugin.*;

//section 1 start *********************************************************
import com.ibm.websphere.objectgrid.ClientClusterContext;
import com.ibm.websphere.objectgrid.CopyMode;
import com.ibm.websphere.objectgrid.ObjectGrid;
import com.ibm.websphere.objectgrid.ObjectGridException;
import com.ibm.websphere.objectgrid.ObjectGridManager;
import com.ibm.websphere.objectgrid.ObjectGridManagerFactory;
import com.ibm.websphere.objectgrid.ObjectGridRuntimeException;
import com.ibm.websphere.objectgrid.ObjectMap;
import com.ibm.websphere.objectgrid.Session;
import com.ibm.websphere.objectgrid.plugins.TransactionCallbackException;
import com.ibm.websphere.objectgrid.security.config.ClientSecurityConfiguration;
import
com.ibm.websphere.objectgrid.security.config.ClientSecurityConfigurationFactory;
import com.ibm.websphere.objectgrid.security.plugins.CredentialGenerator;
import
com.ibm.websphere.objectgrid.security.plugins.builtins.UserPasswordCredentialGene
rator;
//section 1 end *********************************************************
/**
 *
 * The RegexFilterNode sample demonstrates how a Java compute node can be used as
 * a filter node and the use of user defined attributes.
 *
 * <p>
 * The node has two user defined attributes "filterField" and "filterRegex". The
node
 * extract the element value of the first field in the message with name held by
the user attribute
 * "filterField". If the value matches the regular expression held by the user
defined attribute
```

```
 * "filterRegex" the message is propagated to the "out" terminal, otherwise it is
propagated to the "alternate" terminal.
 */
public class RegexFilterNode extends MbJavaComputeNode
{
//section 2 start *********************************************************
      static String username = "xcadmin";
      static String password = "xcadmin";
      static String gridName = "Demo_simple";
      static String mapName  = "Demo_simple";
//    static String mapName  = "Demo_simple.LAT.P";
      static String hostName = "9.42.139.170";

      static ObjectGrid clientGrid = null;
      static synchronized public ObjectGrid getObjectGrid()
      {
            if (clientGrid == null)
            {
                  ClientClusterContext ccc = null;
                  ObjectGridManager ogm =
ObjectGridManagerFactory.getObjectGridManager();
                  ClientSecurityConfiguration clientSC = getAdminClientConfig();
                  try
                        {
                        ccc = ogm.connect(hostName+":2809", clientSC, null);
//                      ccc = ogm.connect(hostName+":2809", null, null);
                        }
                  catch (Throwable e)
                        {
                        System.out.println("--------------------------------------
------------------------");
                        System.out.println("Connection failure");
                        System.out.println("--------------------------------------
------------------------");
                        e.printStackTrace();
                        }
                  if (ccc != null)
                        {
                        try
                              {
                              clientGrid = ogm.getObjectGrid(ccc, gridName);
                              }
                        catch (ObjectGridRuntimeException ogre)
                              {
                              System.out.println("--------------------------------
------------------------------");
                              System.out.println("Object Grid failure");
                              System.out.println("--------------------------------
------------------------------");
                              ogre.printStackTrace();
                              }
                        }
            }
            return clientGrid;
      }

      public static ClientSecurityConfiguration getAdminClientConfig()
      {
            ClientSecurityConfiguration clientSC =
ClientSecurityConfigurationFactory.getClientSecurityConfiguration();
            clientSC.setSecurityEnabled(true);
            CredentialGenerator credGen = new
UserPasswordCredentialGenerator(username, password);
            clientSC.setCredentialGenerator(credGen);
            return clientSC;
      }
//section 2 end *********************************************************

  // The user defined attribute that holds the regular expression.
  private final static String FILTER_REGEX_ATTRIBUTE_NAME = "filterRegex";
  // The user defined attribute that holds the field to match on.
  private final static String FILTER_FIELD_ATTRIBUTE_NAME = "filterField";

  // The regular expression pattern
  private Pattern regex;
  // The XPath expression used to extract the element value.
  private String xpathExpression;

  /* (non-Javadoc)
   * @see
com.ibm.broker.javacompute.MbJavaComputeNode#evaluate(com.ibm.broker.plugin.MbMes
sageAssembly)
   */
  public void evaluate(MbMessageAssembly incomingAssembly) throws MbException
  {
    final String methodName = "evaluate";

//section 3 start *********************************************************
      System.out.println("-------------------------------------------------------
-------");
      System.out.println("Simple Grid Test");
      System.out.println("-------------------------------------------------------
-------");
      System.out.println("username     : "+username);
      System.out.println("password     : "+password);
      System.out.println("gridname     : "+gridName);
      System.out.println("mapname      : "+mapName);
      System.out.println("hostname     : "+hostName);
      System.out.println("-------------------------------------------------------
-------");

      if (getObjectGrid() == null)
      {
            System.out.println("-------------------------------------------------
-------------");
            System.out.println("ERROR: unable to connect to objectgrid at
"+hostName);
            System.out.println("-------------------------------------------------
-------------");
            System.exit(1);
      }
      try
      {
      Session sess = clientGrid.getSession();
      ObjectMap map=sess.getMap(mapName);
//    map.setTimeToLive(60);
//    sess.begin();
      Object data = map.get("TestKey01");
      if (data!=null)
      {
            System.out.println("-------------------------------------------------
-------------");
            System.out.println("NOTE that the record with the following key
existed and was replaced: "+data);
            System.out.println("-------------------------------------------------
-------------");
            map.remove("TestKey01");
      }
      map.insert("TestKey01","TestValue01");
//    sess.commit();
      }
      catch (Exception e)
      {
            System.out.println("-------------------------------------------------
-------------");
            System.out.println("ERROR: failure working with objectgrid at
"+hostName);
            System.out.println("-------------------------------------------------
-------------");
            e.printStackTrace();
      }
//section 3 end *********************************************************

    try
    {
      // First use the XPath expression to extract the field value to match on.
      String fieldValue =
(String)incomingAssembly.getMessage().evaluateXPath(getXPathExpression());

      // Create the matcher from the regex pattern and the field value.
      Matcher matcher = getRegexPattern().matcher(fieldValue);

      // If the field value matches the regex then propagate to "out"
      if(matcher.matches())
      {
        getOutputTerminal("out").propagate(incomingAssembly);
      }
      // Otherwise propagate to "alternate"
      else
      {
        getOutputTerminal("alternate").propagate(incomingAssembly);
      }
    }
    catch (PatternSyntaxException pse)
    {
      // The regex provided by the user is invalid so log the error.
      MbService.logError(this,
                         methodName,
                         RegexFilterNodeMessages.MESSAGE_SOURCE,
                         RegexFilterNodeMessages.INVALID_REGEX,
                         "Invalid regex",
                         new String[] { getRegexPattern().toString() });
    }
  }


  /**
   * Returns the XPath expression to extract the fields value. The expression is
   * created the first time it is required, based on the value of user defined
   * attribute "filterField".
   *
   * @return The XPath expression
   */
  private String getXPathExpression()
  {

    //  Only create is necessary
    if(xpathExpression == null)
    {
      // First get the value of user defined attribute.
      String fieldValue =
(String)getUserDefinedAttribute(FILTER_FIELD_ATTRIBUTE_NAME);
      // The XPath string function automatically convert the value to a string.
      xpathExpression = "string(//"+fieldValue+")";
    }
    return xpathExpression;
  }

  /**
   * Returns a Pattern object instance for regular expression returned by user
defined
   * attribute "filterRegex". The object is create the first time the pattern is
required.
   *
   * @return The Pattern object
   */
  private Pattern getRegexPattern()
  {
    // Only create is necessary
    if(regex == null)
    {
      // Compile the user defined attribute into a Pattern object.
      regex =
Pattern.compile((String)getUserDefinedAttribute(FILTER_REGEX_ATTRIBUTE_NAME));
    }

    return regex;
  }


  /**
   * The class is the ResourceBundle containing all the messages for this
example.
   */
  public static class RegexFilterNodeMessages extends ListResourceBundle
  {
    public static final String MESSAGE_SOURCE =
RegexFilterNodeMessages.class.getName();
```

```
    public static final String INVALID_REGEX = "INVALID_REGEX";

    private Object[][] messages  = {{INVALID_REGEX, "%1 is not a valid regular
expression." }};

    /* (non-Javadoc)
     * @see java.util.ListResourceBundle#getContents()
     */
    public Object[][] getContents()
    {
      return messages;
    }
  }
}
```

Appendix C: The JavaCompute node results in console.txt

```
--------------------------------------------------------------
Simple Grid Test
--------------------------------------------------------------
username     : xcadmin
password     : xcadmin
gridname     : Demo_simple
mapname      : Demo_simple
hostname     : 9.42.139.170
--------------------------------------------------------------
[8/12/11 9:25:37:109 EDT] 64f264f2 RuntimeInfo   I CWOBJ0903I: The internal
version of WebSphere eXtreme Scale is v4.3.0 (7.1.0.3) [cf31124.67080].
[8/12/11 9:25:37:421 EDT] 64f264f2 DefaultDepend I CWOBJ2020I: Client properties
are: preferLocalJVM=true, preferLocalHost=true, preferZones=null,
features=[XSSYSTEM], bootStrapListShuffel=true.
[8/12/11 9:25:37:421 EDT] 64f264f2 DefaultDepend I CWOBJ0063I: The
com.ibm.CORBA.iiop.NoLocalCopies property was not configured. The
com.ibm.CORBA.iiop.NoLocalCopies property is being set to true.
[8/12/11 9:25:37:421 EDT] 64f264f2 DefaultDepend I CWOBJ0063I: The
com.ibm.CORBA.FragmentTimeout property was not configured. The
com.ibm.CORBA.FragmentTimeout property is being set to 30.
[8/12/11 9:25:37:421 EDT] 64f264f2 DefaultDepend I CWOBJ0063I: The
com.ibm.CORBA.FragmentSize property was not configured. The
com.ibm.CORBA.FragmentSize property is being set to 0.
[8/12/11 9:25:37:421 EDT] 64f264f2 DefaultDepend I CWOBJ0063I: The
com.ibm.CORBA.RequestTimeout property was not configured. The
com.ibm.CORBA.RequestTimeout property is being set to 30.
[8/12/11 9:25:37:421 EDT] 64f264f2 DefaultDepend I CWOBJ0063I: The
com.ibm.CORBA.ConnectTimeout property was not configured. The
com.ibm.CORBA.ConnectTimeout property is being set to 10.
[8/12/11 9:25:37:421 EDT] 64f264f2 DefaultDepend I CWOBJ0063I: The
com.ibm.CORBA.NoLocalInterceptors property was not configured. The
com.ibm.CORBA.NoLocalInterceptors property is being set to true.
[8/12/11 9:25:37:421 EDT] 64f264f2 DefaultDepend I CWOBJ0063I: The
com.ibm.CORBA.ThreadPool.MaximumSize property was not configured. The
com.ibm.CORBA.ThreadPool.MaximumSize property is being set to 256.
[8/12/11 9:25:37:421 EDT] 64f264f2 DefaultDepend I CWOBJ0063I: The
com.ibm.CORBA.MinOpenConnections property was not configured. The
com.ibm.CORBA.MinOpenConnections property is being set to 1024.
[8/12/11 9:25:37:421 EDT] 64f264f2 DefaultDepend I CWOBJ0063I: The
com.ibm.CORBA.ConnectionMultiplicity property was not configured. The
com.ibm.CORBA.ConnectionMultiplicity property is being set to 1.
[8/12/11 9:25:37:421 EDT] 64f264f2 DefaultDepend I CWOBJ0063I: The
com.ibm.CORBA.ThreadPool.IsGrowable property was not configured. The
com.ibm.CORBA.ThreadPool.IsGrowable property is being set to false.
[8/12/11 9:25:37:421 EDT] 64f264f2 DefaultDepend I CWOBJ0063I: The
com.ibm.CORBA.LocateRequestTimeout property was not configured. The
com.ibm.CORBA.LocateRequestTimeout property is being set to 10.
[8/12/11 9:25:37:421 EDT] 64f264f2 DefaultDepend I CWOBJ0063I: The
com.ibm.CORBA.ThreadPool.MinimumSize property was not configured. The
com.ibm.CORBA.ThreadPool.MinimumSize property is being set to 256.
[8/12/11 9:25:37:421 EDT] 64f264f2 DefaultDepend I CWOBJ0063I: The
com.ibm.CORBA.MaxOpenConnections property was not configured. The
com.ibm.CORBA.MaxOpenConnections property is being set to 1024.
```

```
[8/12/11 9:25:37:421 EDT] 64f264f2 DefaultDepend I CWOBJ0063I: The
com.ibm.CORBA.ServerSocketQueueDepth property was not configured. The
com.ibm.CORBA.ServerSocketQueueDepth property is being set to 1024.
[8/12/11 9:25:37:625 EDT] 64f264f2 IBMOrbDepende I CWOBJ0917I: Client ORB is
listening on host and port 9.65.200.97:2781.
[8/12/11 9:25:37:625 EDT] 64f264f2 IBMOrbDepende I CWOBJ0915I: ORB version used
is IBM Java ORB build orb60-20100326.00.
[8/12/11 9:25:37:625 EDT] 64f264f2 IBMOrbDepende I CWOBJ0062I: ORB property
com.ibm.CORBA.Debug=<null>.
[8/12/11 9:25:37:625 EDT] 64f264f2 IBMOrbDepende I CWOBJ0062I: ORB property
com.ibm.CORBA.EnableServerKeepAlive=<null>.
[8/12/11 9:25:37:625 EDT] 64f264f2 IBMOrbDepende I CWOBJ0062I: ORB property
com.ibm.CORBA.CodebaseURLEnabled=<null>.
[8/12/11 9:25:37:625 EDT] 64f264f2 IBMOrbDepende I CWOBJ0062I: ORB property
com.ibm.CORBA.SINOClient=<null>.
[8/12/11 9:25:37:625 EDT] 64f264f2 IBMOrbDepende I CWOBJ0062I: ORB property
com.ibm.CORBA.MaxGIOPMinor=<null>.
[8/12/11 9:25:37:625 EDT] 64f264f2 IBMOrbDepende I CWOBJ0062I: ORB property
com.ibm.CORBA.BootstrapHost=<null>.
[8/12/11 9:25:37:625 EDT] 64f264f2 IBMOrbDepende I CWOBJ0062I: ORB property
com.ibm.CORBA.BootstrapPort=<null>.
[8/12/11 9:25:37:625 EDT] 64f264f2 IBMOrbDepende I CWOBJ0062I: ORB property
com.ibm.CORBA.LocalHost=<null>.
[8/12/11 9:25:37:625 EDT] 64f264f2 IBMOrbDepende I CWOBJ0062I: ORB property
com.ibm.CORBA.ListenerPort=<null>.
[8/12/11 9:25:37:625 EDT] 64f264f2 IBMOrbDepende I CWOBJ0062I: ORB property
com.ibm.CORBA.ORBCharEncoding=<null>.
[8/12/11 9:25:37:625 EDT] 64f264f2 IBMOrbDepende I CWOBJ0062I: ORB property
com.ibm.CORBA.enableClientCallbacks=<null>.
[8/12/11 9:25:37:625 EDT] 64f264f2 IBMOrbDepende I CWOBJ0062I: ORB property
com.ibm.CORBA.enableLocateRequest=<null>.
[8/12/11 9:25:37:625 EDT] 64f264f2 IBMOrbDepende I CWOBJ0062I: ORB property
com.ibm.CORBA.NoLocalInterceptors=true.
[8/12/11 9:25:37:625 EDT] 64f264f2 IBMOrbDepende I CWOBJ0062I: ORB property
com.ibm.CORBA.BufferSize=<null>.
[8/12/11 9:25:37:625 EDT] 64f264f2 IBMOrbDepende I CWOBJ0062I: ORB property
com.ibm.CORBA.MaxBufferPoolSize=<null>.
[8/12/11 9:25:37:625 EDT] 64f264f2 IBMOrbDepende I CWOBJ0062I: ORB property
com.ibm.CORBA.LargeDataBufferSize=<null>.
[8/12/11 9:25:37:625 EDT] 64f264f2 IBMOrbDepende I CWOBJ0062I: ORB property
com.ibm.CORBA.MaxLargeDataBufferPoolSize=<null>.
[8/12/11 9:25:37:625 EDT] 64f264f2 IBMOrbDepende I CWOBJ0062I: ORB property
com.ibm.CORBA.requestRetriesCount=<null>.
[8/12/11 9:25:37:625 EDT] 64f264f2 IBMOrbDepende I CWOBJ0062I: ORB property
com.ibm.CORBA.requestRetriesDelay=<null>.
[8/12/11 9:25:37:625 EDT] 64f264f2 IBMOrbDepende I CWOBJ0062I: ORB property
com.ibm.CORBA.RequestTimeout=30.
[8/12/11 9:25:37:625 EDT] 64f264f2 IBMOrbDepende I CWOBJ0062I: ORB property
com.ibm.CORBA.LocateRequestTimeout=10.
[8/12/11 9:25:37:625 EDT] 64f264f2 IBMOrbDepende I CWOBJ0062I: ORB property
com.ibm.CORBA.FragmentTimeout=30.
[8/12/11 9:25:37:625 EDT] 64f264f2 IBMOrbDepende I CWOBJ0062I: ORB property
com.ibm.CORBA.ConnectTimeout=10.
[8/12/11 9:25:37:625 EDT] 64f264f2 IBMOrbDepende I CWOBJ0062I: ORB property
org.omg.CORBA.ORBServerId=<null>.
[8/12/11 9:25:37:625 EDT] 64f264f2 IBMOrbDepende I CWOBJ0062I: ORB property
com.ibm.CORBA.ConnectionMultiplicity=1.
```

```
[8/12/11 9:25:37:625 EDT] 64f264f2 IBMOrbDepende I CWOBJ0062I: ORB property
com.ibm.CORBA.MaxOpenConnections=1024.
[8/12/11 9:25:37:640 EDT] 64f264f2 IBMOrbDepende I CWOBJ0062I: ORB property
com.ibm.CORBA.MinOpenConnections=1024.
[8/12/11 9:25:37:640 EDT] 64f264f2 IBMOrbDepende I CWOBJ0062I: ORB property
com.ibm.CORBA.ServerSocketQueueDepth=1024.
[8/12/11 9:25:37:640 EDT] 64f264f2 IBMOrbDepende I CWOBJ0062I: ORB property
com.ibm.CORBA.GIOPAddressingDisposition=<null>.
[8/12/11 9:25:37:640 EDT] 64f264f2 IBMOrbDepende I CWOBJ0062I: ORB property
com.ibm.CORBA.SendingContextRunTimeSupported=<null>.
[8/12/11 9:25:37:640 EDT] 64f264f2 IBMOrbDepende I CWOBJ0062I: ORB property
com.ibm.CORBA.ThreadPool.IsGrowable=false.
[8/12/11 9:25:37:640 EDT] 64f264f2 IBMOrbDepende I CWOBJ0062I: ORB property
com.ibm.CORBA.ThreadPool.MaximumSize=256.
[8/12/11 9:25:37:640 EDT] 64f264f2 IBMOrbDepende I CWOBJ0062I: ORB property
com.ibm.CORBA.ThreadPool.MinimumSize=256.
[8/12/11 9:25:37:640 EDT] 64f264f2 IBMOrbDepende I CWOBJ0062I: ORB property
com.ibm.CORBA.ThreadPool.InactivityTimeout=<null>.
[8/12/11 9:25:37:640 EDT] 64f264f2 IBMOrbDepende I CWOBJ0062I: ORB property
com.ibm.CORBA.AlwaysUseOMG4796=<null>.
[8/12/11 9:25:37:640 EDT] 64f264f2 IBMOrbDepende I CWOBJ0062I: ORB property
com.ibm.CORBA.UseOMG5689=<null>.
[8/12/11 9:25:37:640 EDT] 64f264f2 IBMOrbDepende I CWOBJ0062I: ORB property
com.ibm.CORBA.TransportMode=<null>.
[8/12/11 9:25:37:640 EDT] 64f264f2 IBMOrbDepende I CWOBJ0062I: ORB property
com.ibm.CORBA.AllowUserInterrupt=<null>.
[8/12/11 9:25:37:640 EDT] 64f264f2 IBMOrbDepende I CWOBJ0062I: ORB property
com.ibm.CORBA.DisableOMG3681=<null>.
[8/12/11 9:25:37:640 EDT] 64f264f2 IBMOrbDepende I CWOBJ0062I: ORB property
org.omg.CORBA.ORBId=<null>.
[8/12/11 9:25:37:640 EDT] 64f264f2 IBMOrbDepende I CWOBJ0062I: ORB property
org.omg.CORBA.ORBListenEndpoints=<null>.
[8/12/11 9:25:37:640 EDT] 64f264f2 IBMOrbDepende I CWOBJ0062I: ORB property
org.omg.CORBA.ORBNoProprietaryActivation=<null>.
[8/12/11 9:25:37:640 EDT] 64f264f2 IBMOrbDepende I CWOBJ0062I: ORB property
com.ibm.CORBA.UseHarmonyClassLibrary=<null>.
[8/12/11 9:25:38:906 EDT] 64f264f2 JvmMemoryUtil I CWOBJ4542I: Basic BackingMap
memory sizing is enabled.
---------------------------------------------------------------
NOTE that the record with the following key existed and was replaced: TestValue01
---------------------------------------------------------------
```