IBM Integration Bus

Global Cache using a DataPower XC10 appliance

June, 2013

Hands-on lab built at product code level Version 9.0.0.0
1. INTRODUCTION TO INTEGRATION BUS GLOBAL CACHE ............................................................3
2. SCENARIO .....................................................................................................................................4
3. PREPARATION FOR THIS LAB GUIDE ....................................................................................6
   3.1 IBM INTEGRATION BUS PREPARATION ...........................................................................7
   3.2 CONNECTIVITY TO A DATAPOWER XC10 APPLIANCE ..............................................8
      3.2.1 Using a Virtual DataPower XC10 appliance ...............................................................9
      3.2.2 Using a physical DataPower XC10 appliance ............................................................16
4. CONFIGURE YOUR ENVIRONMENT .......................................................................................17
   4.1 CREATE A DATA GRID ON THE XC10 APPLIANCE ....................................................17
   4.2 CREATE WXSS SERVER CONFIGURABLE SERVICE .......................................................24
   4.3 CONFIGURE PORTS AND LISTENER FOR THE INTEGRATION NODE ......................26
5. IMPORT THE APPLICATIONS ..................................................................................................27
   5.1 REVIEW THE XC10_REQUEST APPLICATION ..............................................................28
      5.1.1 Modify XC10SaveToCache.java source ..................................................................29
   5.2 REVIEW THE XC10_CUSTOMER APPLICATION ...........................................................31
   5.3 REVIEW THE XC10_RESPONSE APPLICATION ..............................................................32
6. DEPLOY AND TEST THE APPLICATIONS ..............................................................................34
   6.1 SHOW THE CACHE RESOURCE STATISTICS IN INTEGRATION EXPLORER ...............37
   6.2 VIEW THE XC10 DATA GRID/MAP STATISTICS ...........................................................39
This concludes the Global Cache using a DataPower XC10 appliance Lab ...............................41
7. APPENDIX A .............................................................................................................................42
   7.1 USING DATAPOWER XC10 AT V2.5 FIRMWARE ...............................................................42
1. Introduction to Integration Bus Global Cache

WebSphere Message Broker V8.0.0.1 (fix pack 1), provided a new feature called Global Cache. IBM WebSphere Message Broker V8.0.0.2 (fix pack 2) further extended this feature to provide an automatic data revocation feature. IBM Integration Bus V9.0.0.0 provides these features without the need to activate them using the `mqsichangebroker` command.

Global Cache allows a number of Integration Nodes to be integrated for workload balancing. Global Cache provides the ability to have a cache storing information about the requester which would later be used to correlate the replies correctly. Using Global cache gives the flexibility for different Integration Nodes to handle the request and reply parts of an application. Using a global cache allows you to scale up the number of clients, while maintaining predictable response time for each client.

Global Cache uses an embedded version of WebSphere eXtreme Scale, and provides the following functions:

• 'Elastic' In-Memory Data Grid – managing itself to scale out, scale in, failover, failure etc.
• Virtualizes free memory within a grid of Java Virtual machines (JVMs) into a single logical space which is accessible as a partitioned, key addressable space for use by applications
• Provides fault tolerance through replication with self-monitoring and healing
• The space can be scaled out by adding more JVMs while its running without restarting
• Provides a predictable scaling option
• Access to external WebSphere Xtreme Scale data grids

IBM Integration Bus Version 9.0.0.0 contains support for:

1) An embedded WebSphere eXtreme Scale (WXS) grid which can be used as a global cache from within message flows. WXS components are hosted within Integration Server processes and operate with no requirement for additional configuration. The default scope of one cache is across one Integration node (i.e. multiple Integration Servers) but it can be extended to be across multiple Integration nodes. Integration Bus support for embedded WebSphere Extreme Scale grids is covered under the Lab Guide "IBM Integration Bus, Version 9.0.0.0, Embedded Global Cache" from the same series of Lab Guides.

2) Access to WebSphere Xtreme Scale Grids that are running outside of the Integration Node for example a DataPower XC10 appliance.

This lab guide demonstrates a simple setup / configuration of an application using a DataPower XC10 appliance.

This lab guide has been written with the XC10 firmware at V2.1. XC10 Firmware at V2.0 and V2.5 will also work when communicating with IBM Integration Bus V9.0. A small change in the XC10 configuration is required when using XC10 V2.5 firmware. This change is documented in Appendix A of this guide. Some screen captures may appear slightly different when using the V2.5 level of the firmware.
2. Scenario

The XC10 Global cache scenario is based on the Integration Bus embedded Global Cache scenario and demonstrates the configuration required to use the external DataPower appliance.

The following schematic shows the system context of the Global Cache Lab.

There are three message flows in this scenario.

1. **XC10_Request** – this flow receives messages on the `GLOBAL.CACHE.XC10.IN` queue. The application that puts messages on this queue also specifies the ultimate reply queue, which will be set to `GLOBAL.CACHE.XC10.OUT` in this example.

   The Request message flow takes the MessageID and ReplyToQ from the incoming message, and stores them in an XC10 appliance data grid. This is so that the Response message flow will be able to use this data to send the reply back to the specified reply queue, even though Response may be running in a different Integration Server, or Integration Node.

   The Request flow has a user-defined property, which specifies the name of the MQ queue that the retrieveCustomer flow will send the resulting message to. This value is used to populate the ReplyToQ that will be used by retrieveCustomer. This has been done for ease of illustration of the Global Caching component, when switching the demo between multiple Integration Servers and Integration Nodes.
2. retrieveCustomer – this flow takes the input message (passed by Request) and uses the data to retrieve a customer record from the CUSTOMER database. This is done with a simple Compute node. It then sends the output message to the specified Reply queue. This flow is the same flow used for the embedded global cache application and has nothing specific to the XC10 appliance.

3. XC10_Response - runs in both brokers, and reads the GLOBAL.CACHE.RESPONSE.XC10.IN queue. It uses the MessageID to retrieve the required final reply queue name from the global cache, and sends the message to the final output queue.
3. Preparation for this Lab Guide

This lab guide uses IBM Integration Bus and a DataPower XC10 appliance. The following sections outline the preparation for IBM Integration Bus and options for using an external DataPower XC10 appliance.
3.1 IBM Integration Bus preparation

In the pre-built VMware, this section has been done for you. If you have previously done the embedded Global Cache Lab in this series, the tables and queues will also have been set up. You can ignore this section.

To run this lab, unzip the supplied file global_cache.zip into the directory c:\student\global_cache. This will create a subdirectory called global_cache with several further subdirectories.

Create the MQ Queues

In the c:\student\global_cache\install folder, run the command createCacheQueues.bat.

Create the databases

In a Integration Command Console, go to the folder c:\student\global_cache\install. Issue the commands:

- CreateRegularDB
- CreateRegularTables
- SetBrokerSecurityForCaching
- mqsistop IB9NODE
- mqsistart IB9NODE
This lab guide shows you how to configure IBM Integration Bus V9.0 to communicate and use a DataPower XC10 appliance. There are two possible choices of XC10 device when communicating with IBM Integration Bus V9.0:

1) Use a virtual XC10 DataPower appliance.
2) Use a physical XC10 DataPower appliance.

The following sections will outline the necessary pre-req setup when using both of these options.
If you are using a physical DataPower XC10 appliance you can ignore this section and go to the next section entitled “Using a physical DataPower XC10 appliance”.

3.2.1.1 Verify the XC10 appliance ip address

1. Focus your keyboard input to the virtual appliance console by clicking the (black) virtual XC10 console screen. (There may be initialization messages on the screen from the appliance start up process.)

2. Press the <enter key>, a login prompt will appear.

3. login using xcadmin, with a password of xcadmin:

When you are successfully signed on to the appliance you will see something similar to the following:

```
 XC10 appliance console

WebSphere DataPower XC10
2.5.0.0-0.0-CF12345.678901
eth0: addr:192.168.192.132
eth1

CLXSA0014I: Volume 1 mounted successfully.
CLXSA0061I: The catalog server has started.
CLXSA0018I: The data grid configuration service has started.
CLXSA0017I: The xsServer01 container server has started.
CLXSA0082I: The data grid administrative service has started.
CLXSA0083I: The administrative console is starting.
STARTED
```

(none) login: xcadmin
Password: _

(none) login: xcadmin
Password:
Last login: Wed Jul 10 10:16:29 UTC 2013 on tty1
Last login: Wed Jul 10 10:24:55 on tty1
Welcome to WebSphere Datapower XC10 Appliance
Console>
4. At the console prompt type the command:

```
status ethernet-interface eth0
```

The command will respond with the following:

The IP address of the appliance is shown by the "inet addr:" prefix (in the above example 192.168.192.132)

Write down the IP address of the appliance in your environment here:

XC10 IP Address: ________________________________
6. Enter the IP address of the virtual XC10 appliance in the Firefox browser. You will be prompted that the connection is “Untrusted”, click “I understand the Risks” and the “Add Exception” button.

7. Click the “Get Certificate” button.

8. Click the “View” button to make sure the IP address you entered is the XC10 virtual appliance, it will look similar to this: Close the viewer window.
8. Back in the "Add security Exception" window, click "Confirm Security Exception":

9. The Web Administration GUI sign-on screen will appear. Log in using "User name" xcadmin and password xcadmin:
On the main XC10 Web Administration GUI window click "Collective" then "Users".

In the "Users" screen click the green plus symbol (to add a user):

1. Type a User name (user01 in the case below), with a password of "passw0rd" (password with a zero instead of the 'o'), click OK when complete.
3.2.1.3 Define permissions for user

13. When added to the list of users on the virtual appliance, enable the user with "Data cache creation" permission by check marking the box for "Data cache creation":

14. Log out of the Web Administration tool by clicking the "Log Out" link:

<table>
<thead>
<tr>
<th>User</th>
<th>Permissions</th>
<th>User</th>
<th>Inherited from groups</th>
</tr>
</thead>
<tbody>
<tr>
<td>user01</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>User name: user01</td>
<td>Email address: None provided</td>
<td>Password: ******** [edit]</td>
<td>Current status: User has not logged in yet</td>
</tr>
<tr>
<td>User groups: Everyone</td>
<td>Add more...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Data cache creation**
15. Sign on using the user that you just added:

16. Notice the user has the ability to create and monitor a data grid however cannot administer the appliance (i.e., cannot change appliance settings or add users):

17. You are now ready to proceed with the instructions in this Lab Guide. This is the XC10 user that you will use throughout this lab guide, the remainder of this lab guide will refer to this user as userXX.
If you are using a Virtual DataPower XC10 appliance you can ignore this section.

If you are using a physical XC10 device, ask your XC10 administrator for:

1) The IP address of the DataPower XC10 administration interface.
2) An XC10 user called userXX (XX can be a number between 0 and 99 if multiple users are required)
3) Permissions: userXX must have "Data cache creation" and "Appliance monitoring" permissions.

NB: If you need to see how to add a user on a physical appliance the process is exactly the same as documented in the previous section where you add a user to the virtual XC10 appliance (note passwords on your physical device may be different).
4. Configure your environment

This section and remainder of this lab guide is relevant to both XC10 options (physical and virtual).

4.1 Create a data grid on the XC10 appliance

The XC10 appliance provides a web based configuration tool. You will now use this web based tool to create a simple data grid to store information used in this lab guide.

18. Point a browser at the IP address of your XC10 Appliance.

The Configuration tool login screen will appear. Sign on with your XC10 user credentials. (you will either have been given this user if using a physical appliance – or added a user in the previous section if you are using the supplied virtual appliance - for the purposes of illustration throughout this lab we will use userXX)
19. On the Home page select "Simple Data Grid" from the "Data Grid" drop down:

20. On the "Simple Data Grids" screen:

21. Enter a name for the grid, and click OK (make the grid name specific to the user you are using by replacing XX with the numeric digits applicable to your user for example user01datagrid if you are user01)

22. On the next pop up click the "Tasks View":

The creation of a data grid can take several minutes to complete. Click the Tasks View button to monitor the creation in the tasks view, or click the Stay Here button to remain on the current page.
23. In the "Tasks" view you will see that your request has been scheduled and is "running". On a light use system a data grid will take 1 or 2 minutes to create on the appliance:

24. When complete the "Tasks" view will update with the completion status for the request. Ensure your grid is created successfully (status: Success in the task view):
25. Navigate back to the Simple Data Grid menu and click on the link to the grid you have just created. Make a note of the Catalog Server (IP address and port): ____________________________
Click "Show advanced attributes", scroll to the bottom of the advanced attributes for the data grid you created. In the "Time to live eviction" section, set 1) the "Evictor" for the map to "Last access time" and 2) "Evict data from time to live maps after" to "300" seconds.

When an XC10 data grid is created the default eviction policy is set so that data is kept until the grid is restarted. Changing the Time to Live eviction at this level means that data not accessed for 5 minutes (300 seconds) or longer will be automatically removed from data grid.

Click "Apply changes" and then OK to confirm the changes (and then "Tasks View"). Note that applying this action will restart the data grid (any existing data in the data grid is removed on a restart).
In the "Task View" you will see the "eviction update" with a status (queued or running). The screen will automatically update when the task has completed (the status will change to "success").
28. The XC10 configuration tool also provides summary statistics on data grid usage. Click on "Monitor" and choose "Data Grid Detail Reports". Note that there will have been no usage on your grid at this stage:

- Current summary over last 5 seconds
- Used Capacity (B): 0.00
- Number of cache entries: 0
- Average Throughput: 0.00
- Average Transaction Time: 0.00

29. Your XC10 data grid is now ready to be used. The next configuration task is to make IIB aware of the data grid.

Move to the next section "Create WXSServer configurable service".
4.2 Create WXSServer configurable service

Details of the data grid and connectivity details for the XC10 server are stored in an Integration Bus configurable service of type "WXSServer". Note this configurable service defines the grid that you will be using (not just the connection specific details of the XC10 appliance). Since multiple data grids can exist on an XC10 appliance more than one definition of type "WXSServer" can exist in an environment (with the same XC10 connection details).

For the purposes of this lab guide, this configuration will be done using the Integration Explorer plugin (in WebSphere MQ Explorer), the `mqsicreateconfigurableservice` command can be used as an alternative.

You will now use the Integration Explorer to define a configurable service that the Integration Node can use to connect to the XC10 data grid:

30. In Integration Explorer, expand IB9NODE and Right click on "Configurable Services", choose "New" then "Configurable Service…" from the options:
31. In the Wizard that opens change the "Type" to "WXSServer" and enter:
1. the IP and port details of the XC10 catalog server (you made a note of it earlier)
2. the name of the data grid ("userXXdatagrid") (replace userXX with your details)
3. Name the service "CS_XC10DATAGRID", (case is important since we will be using this value in the Java compute nodes later in the lab)

Click Finish to save:

The Configurable service CS_XC10DATAGRID will appear in the list of configurable services for the IB9NODE integration node:
4.3 Configure ports and listener for the Integration Node

Integration Nodes use the embedded cache to connect to the XC10 DataPower appliance. The embedded cache requires configuring even if you decide not to use the embedded Global Cache.

The default port range used by the default Integration Node for the embedded eXtreme Scale component is 2800-2819. In the event that this port range conflicts with any other applications, you can change this port range using the Integration Explorer.

In the case of the pre-built system which includes WAS and Information Server, this is necessary to avoid a port conflict.

1. In Integration Explorer, right click on IB9NODE, and select Properties. Select Global Cache.
2. Set Cache policy to "Default – single broker cache managed by the broker"
   Change the port range to 3840-3859.
3. Set the Listener host name to "BETAWORKS-ESB01" (the host name of your VM machine – check this value by entering "hostname" in a command window).
   Click Apply, then OK.
3. Stop and restart the Integration Node.
5. Import the applications

The application that you will use to investigate IBM Integration Bus support of the DataPower XC10 Appliance is provided for you. In this section, you will import the application, and investigate certain aspects of the flow logic. The application that is used to demonstrate IBM Integration Bus use of the XC10 DataPower appliance for caching is very similar to the application used with the embedded global cache. The following sections will highlight the differences when using the XC10 appliance.

4. In the Integration Toolkit, import the Project Interchange file at the following location:

   c:\student\global_cache\resources\ XC10_Start.zip

5. Explore the items that have been defined in the Application Development navigator.

   Things to note:

   There are three applications, each containing one message flow. The application has been organized in this way in order that each application (and therefore message flow) can be deployed independently to separate Integration Servers, or to separate Integration Nodes.

   The XC10_Request and XC10_Response applications have a reference to the WXSJava_Lib library. The WXSJava_Lib library contains four java compute nodes, two of these are relevant to the XC10 cache scenario: One for writing information to the XC10 cache, and one for reading from it.

   The XC10_Customer application has no reference. The retrieveCustomer message flow is independent, and uses a simple Compute node to read a DB2 database.
5.1 Review the XC10_Request application

6. Open the **XC10_Request** message flow:

This message flow reads an MQ message from queue **GLOBAL.CACHE.XC10.IN**, and uses a java compute node to store the MQ MessageID and Reply Queue Name (required to correlate the final MQ response message) into the cache. The message is then sent to the **XC10_retrieveCustomer** message flow by writing it to the **GLOBAL.CACHE.XC10.CUSTOMER.IN** queue.
7. In the flow editor, click "User defined properties". You will see that this flow has a user-defined property called "ResponseQueue". This UDP is used to determine whether the output from the retrieveCustomer flow is sent to a local queue, or to a remote queue (for the purpose of demonstrating caching across multiple brokers). The default value is "GLOBAL.CACHE.XC10.RESPONSE.IN", which is a local MQ queue.

5.1.1 Modify XC10SaveToCache.java source

8. Double-click on the "XC10SaveToCache.java" source file in WXSJava_Lib to see how the Java logic writes information to the cache. There are three key sections of java code:

This section of code retrieves the MQ MsgId and ReplyToQ from the MQMD (same code is in the default global cache example):

```java
// Get the original MsgId and ReplyToQ of the incoming request
MbElement rootEl = outAssembly.getMessage().getRootElement();
MbElement replyToQEl = rootEl.getFirstElementByPath("/MQMD/ReplyToQ");
String replyToQ = replyToQEl.getValueAsString();
MbElement msgIdEl = rootEl.getFirstElementByPath("/MQMD/MsgId");
String msgId = msgIdEl.getValueAsString();
```
9. This section of code (unique to using the XC10 appliance) saves this information in the global cache:

```java
/*
 * We're about to overwrite the original ReplyToQ of this message.
 * So, write it to the XC10 cache for safe keeping. This will be
 * used subsequently by the XC10 Response message flow to send the
 * response back to the original reply queue.
 */

MbGlobalMap xc10Map = MbGlobalMap.getGlobalMap("userXXdatagrid","CS_XC10DATAGRID");
xc10Map.put(msgId, replyToQ);
```

Action: Overtype the value "userXXdatagrid" to reflect the details of the data grid you created on the XC10.

10. A Note on data grids and Maps: an XC10 data grid can contain multiple "Maps". When you create a data grid, a single map (with the same name as the grid name – in our case "userXXdatagrid") is created by default. It is this map name (known as a GlobalMap object) that getGlobalMap requires.

It is possible to dynamically create additional maps in a data grid by the WSX client attempting to access a specifically named map in this field. For example a map name of "userXXdatagrid.LAT" specified in the code would:

a) dynamically create a map within the data grid if it didn't exist
b) since the name consists of "<anyname>.LAT" a default data eviction policy (this is when data will be removed from the map) would be based on "Last Access Time" of the entry
c) a default time to live (TTL) of 1 hour. (this would mean data not accessed for 1 hour would be automatically deleted from the map.

More information on XC10 dynamic map configuration options can be found at the following URL:


11. Finally, this section of code reads the value of a user-defined property "ResponseQueue". The value of this user-defined property determines whether the reply queue of the retrieveCustomer flow will be a local or remote queue.

```java
responseQueue = getUserDefinedAttribute("ResponseQueue").toString();
outMessage.getRootElement().getFirstElementByPath("/MQMD/ReplyToQ").setValue(responseQueue + instance);
```

12. Save the java code and exit the java editor when finished, and close the Request message flow.
5.2 Review the XC10_Customer application

13. Now open the retrieveCustomer message flow. On the MQ Input node, the Message Parsing domain has been set to XMLNSC. This is because the flow needs access to the input data (CustomerID) to retrieve a record from the CUSTOMER table.

14. Open the getCustomer Compute node. The ESQL code will retrieve a row from the CUSTOMER table, putting it temporarily in Environment.Variables. The FirstName and LastName fields are then stored in the output message.

BEGIN
-- CALL CopyMessageHeaders();
SET OutputRoot = InputRoot;
SET OutputRoot.XMLNSC.CUSTOMER = NULL;
-- populate the environment with passenger info from the database
SET Environment.Variables =
    THE (SELECT T.* FROM Database.CUSTOMER AS T
         WHERE T.CUSTOMERID = InputRoot.XMLNSC.CUSTOMER.CUSTOMERID);

-- populate the output message with info from the database query
CREATE FIELD OutputRoot.XMLNSC.CUSTOMER;
DECLARE outpass REFERENCE TO OutputRoot.XMLNSC.CUSTOMER;
SET outpass.FirstName = Environment.Variables.FIRSTNAME;
SET outpass.LastName = Environment.Variables.LASTNAME;
RETURN TRUE;
END;
5.3 Review the XC10_Response application

15. Now open the Response message flow. First, note that the MQ input node reads the queue `GLOBAL.CACHE.XC10.RESPONSE.IN`.

16. Open the `XC10RetrieveFromCache` java node. The following are the important parts of the java code.

```java
String msgId = inAssembly.getMessage().getRootElement().getFirstElementByPath("/MQMD/CorrelId").getValueAsString();
```

17. The following code uses the MessageID to retrieve the name of the reply queue from the Global Cache.

```java
String replyToQ = (String)globalMap.get(msgId);
```

Note: the MessageID retrieved is NOT deleted from the XC10 data grid in this scenario using the code. This data will be "cleaned up" using the data eviction policy set for the data grid.

---

The diagram shows the flow of data from the MQ input node to the `XC10RetrieveFromCache` node, and then to the MQ Reply node.

---

```
String msgId = inAssembly.getMessage().getRootElement().getFirstElementByPath("/MQMD/CorrelId").getValueAsString();
```

Action: Overtype the data grid value "userXXdatagrid" to reflect the details of the data grid you created on the XC10.

// Now we can restore the original ReplyToQ by looking it up in the cache.
```
MbGlobalMap xc10Map = MbGlobalMap.getGlobalMap("userXXdatagrid","CS_XC10DATAGRID");
String replyToQ = (String)globalMap.get(msgId);
```

Note: The MessageID retrieved is NOT deleted from the XC10 data grid in this scenario using the code. This data will be "cleaned up" using the data eviction policy set for the data grid.
18. This section sets the ReplyToQ field in the output message, and resets the value of the MessageID to the original value.

```java
// Set the ReplyToQ field
outMessage.getRootElement().getFirstElementByPath("/MQMD/ReplyToQ").setValue(replyToQ);

// Set the MsgId back to what it was before
outMessage.getRootElement().getFirstElementByPath("/MQMD/MsgId").setValue(getBytes(msgId));
```

19. Save and close the RetrieveFromCache.java code.
6. Deploy and test the applications

This part of the lab will test the caching applications, with all applications deployed to a single Integration Node. To demonstrate the global cache, each of the three applications will be deployed into a separate Integration Servers, as follows:

1. Default – XC10_Customer
2. WXS_Request_Server – XC10_Request
3. WXS_Response_Server – XC10_Response

Note: If you have previously deployed the embedded global cache applications into the Integration Servers please delete these applications from the Integration Servers so that you have a “clean” working environment for this XC10 lab.

4. Create two new Integration Servers, WXS_Request_Server and WXS_Response_Server. (Right-click the IB9NODE, and select “New Integration Server”.)
5. Deploy the applications as shown above. Drag and drop the application into the required Integration Server.
6. When all three applications are deployed, the Integration Node should look something like this:

7. Open RFHUtil, and open the file named `c:\student\global_cache\data\Customer0001.xml`.
This data contains a CustomerID with a value of 0001. The CustomerID will be used to retrieve customer data from the SUBREG1 database, by the XC10_Customer application. To ensure the request/reply scenario works, you must specify the name of the reply queue, which is the name of the queue where the final output will appear (i.e., the Response application will write to this queue).

On the MQMD tab, set the "Reply to Queue" to `GLOBAL.CACHE.XC10.OUT`.

On the `Main` tab, write the message to the input queue `GLOBAL.CACHE.XC10.IN`. 
3. Both the graphical and tabular views will open. On the tabular view, click the GlobalCache tab. You will see that the MapWrites count for the XC10 data grid that you are writing to will increment (try using writing a few more messages using RFHUTIL):

And the corresponding Resource statistics Graphical view:

4. Performing the same tasks on the WXS_Response_Server will show the GlobalCache Reads count being incremented.
6.2 View the XC10 data grid/map statistics.

When using the XC10 data grid for caching the mqsicacheadmin command is not useful since the command shows statistics for the use of the embedded global cache. The web based XC10 configuration tool provides monitoring statistics reports for the data grids on your XC10 device.

1. Sign on to the XC10 using the userid/password you have been using for the XC10.

   Click the "Monitor" link and choose > "Data Grid Detail Reports",
   In the left navigation expand userXXdatagrid > Data grid maps
   and click on "userXXdatagrid" to show the detailed report for your data grid:

2. In the Integration Explorer have the Resource statistics views open.

3. Using RFHUTIL write lots of messages (>30) to the GLOBAL.CACHE.XC10.IN queue. This will cause the XC10 application to write entries to the XC10 data grid. (ensure all the iterations of the application result in messages being written to the queue GLOBAL.CACHE.XC10.OUT)
4. On the web-based XC10 configuration tool, you will begin to see the "Detailed Report" for your map fill with data (click the name of the report to see the updates):

5. Switch to the Integration Explorer and review the WXS_Request_Server statistics. You will also see that the statistics for Mapwrites for "userXXdatagrid" will have increased:

6. Now take a 5-minute break (no really you've earned it, you got this far :o)

Remember when we set up the data grid – we set the "Time to Live" in the data grid (Advanced attributes section). We set the value at 300 seconds (5 minutes) with an eviction policy of "Last Access Time" meaning that data not accessed for more than 5 minutes will be deleted from the map by the XC10.
When you come back from your break (shouldn’t be hard for it to last 5 minutes or longer ;-) Refresh the Detailed Map Report on the XC10 web admin tool (sign on again if it prompts you). You will see that the data entries and Chart details in the report have changed showing that the grid contains no data:

Write a few more messages to the GLOBAL.CACHE.XC10.IN queue using RFHUTIL and refresh the detailed report again to see that the cache is still functioning correctly.

This concludes the Global Cache using a DataPower XC10 appliance Lab.
If your XC10 firmware is at V2.5 there is a small change to the configuration settings that is required before IBM Integration Bus V9.0 can communicate with your XC10 appliance. Changing this setting requires a restart of the XC10 appliance services which will result in loss of all cached data in all Data Grids on the appliance. Ensure nobody else is actively using the appliance before you attempt this change.

This section of the lab guide assumes that you have already done “Connectivity to a DataPower XC10 appliance” (ie you know the details of the XC10 appliance you are using and have administrator access to the appliance).
2. On the main menu, click "Appliance" then "Settings".

3. Click "Firmware" to verify the version.

4. If you are at version 2.5, you will need to configure the "Transport and storage mode" (in Collective settings) in order for IBM Integration Bus to communicate with the XC10 appliance.

The current firmware version is IBM WebSphere DataPower XC10 Appliance 2.5.0.0-d21324.1313643.
5. Click “Collective” then “settings”:

6. Change the “Transport and storage mode” to “ORB with heap memory (deprecated)”:

7. Click “Apply” when the message appears to warn about restarting the machine:
8. Click OK when prompted.

9. The GUI will redirect you to the task view where you will see that the appliance will be restarted. After a few minutes the task page will indicate that the monitoring user interface will be lost as a result of the configuration change.
10. Reload the web interface pages until you are prompted with the sign on page again (this may take a few minutes whilst the web UI interface on the appliance is restarting).

11. Sign on using xcadmin/xcadmin:

12. Click “Collective”, then “settings” again and ensure that the Transport and storage mode is set to “ORB with heap memory (deprecated)”. You are now ready to connect IBM Integration Bus with your XC10 appliance at V2.5 level.