L13

IBM Integration Bus

Accessing the Global Cache using a Mapping Node

Featuring:

The REST API tools for IIB
Map node using Cache Put and Get key-value pairs

January 2016
Hands-on lab built at product
Version 10.0.0.3

1. Objectives

In this lab, you will implement a REST operation to retrieve data from the Global Cache.

To allow the lab to be completed in the appropriate time, we have provided a REST application that has already implemented the operation to read a database table, and to load the data from the table into the global cache. You will perform the following tasks:

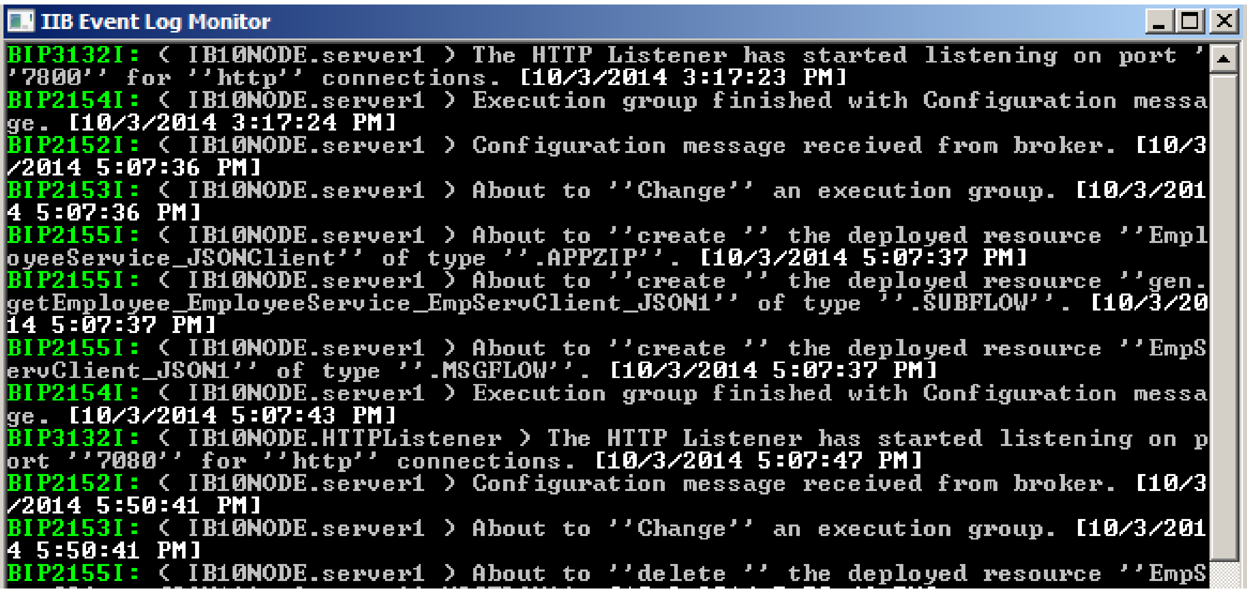
- Import the partially-built REST application
- Investigate the supplied "Load cache" operation
- Configure the IIB node for Global Cache operations
- Deploy and test the supplied "Load Cache" operation
- Implement and test a new operation to retrieve data from the Global Cache

2. Prepare the IIB Node

2.1 Open the Windows Log Monitor for IIB

A useful tool for IIB development on Windows is the IIB Log Viewer. This tool continuously monitors the Windows Event Log, and all messages from the log are displayed immediately.

From the Start menu, click IIB Event Log Monitor. The Monitor will open; it is useful to have this always open in the background.



```
BIP3132I: ( IB10NODE.server1 ) The HTTP Listener has started listening on port '
'7800'' for ''http'' connections. [10/3/2014 3:17:23 PM]
BIP2154I: ( IB10NODE.server1 ) Execution group finished with Configuration messa
ge. [10/3/2014 3:17:24 PM]
BIP2152I: ( IB10NODE.server1 ) Configuration message received from broker. [10/3
/2014 5:07:36 PM]
BIP2153I: ( IB10NODE.server1 ) About to ''Change'' an execution group. [10/3/201
4 5:07:36 PM]
BIP2155I: ( IB10NODE.server1 ) About to ''create '' the deployed resource ''Empl
oyeeService_JSONClient'' of type ''.APPZIP''. [10/3/2014 5:07:37 PM]
BIP2155I: ( IB10NODE.server1 ) About to ''create '' the deployed resource ''gen.
getEmployee_EmployeeService_EmpServClient_JSON1'' of type ''.SUBFLOW''. [10/3/20
14 5:07:37 PM]
BIP2155I: ( IB10NODE.server1 ) About to ''create '' the deployed resource ''EmpS
ervClient_JSON1'' of type ''.MSGFLOW''. [10/3/2014 5:07:37 PM]
BIP2154I: ( IB10NODE.server1 ) Execution group finished with Configuration messa
ge. [10/3/2014 5:07:43 PM]
BIP3132I: ( IB10NODE.HTTPListener ) The HTTP Listener has started listening on p
ort ''7080'' for ''http'' connections. [10/3/2014 5:07:47 PM]
BIP2152I: ( IB10NODE.server1 ) Configuration message received from broker. [10/3
/2014 5:50:41 PM]
BIP2153I: ( IB10NODE.server1 ) About to ''Change'' an execution group. [10/3/201
4 5:50:41 PM]
BIP2155I: ( IB10NODE.server1 ) About to ''delete '' the deployed resource ''EmpS
```

This tool is not shipped as part of the IIB product; please contact us directly if you would like a copy.

2.2 Create and configure a new node for REST

Login to Windows as the user "iibuser", password = "passw0rd".

Start the IIB Toolkit from the Start menu.

This lab uses the REST API function in IBM Integration Bus version 10. The IIB support for the REST API requires some special configuration for the IIB node and server.

1. Ensure that TESTNODE_iibuser is started.

2. Enable Cross-Origin Resource Scripting for REST. This is required when testing with the SwaggerUI test tool. See http://en.wikipedia.org/wiki/Cross-origin_resource_sharing for further information.

(Helpful hint - the VM keyboard may be set to UK English. if you cannot find the "\" with your keyboard, use "cd .." to move the a higher-level folder in a DOS window).

In an IIB Command Console (shortcut on the Start menu), run the command:

```
mqsichangeproperties TESTNODE_iibuser
      -e default
      -o HTTPConnector
      -n corsEnabled -v true
```

2.3 Configure the new node for Global Cache

1. In an IIB Command Console, run the commands

   mqsistop TESTNODE_iibuser

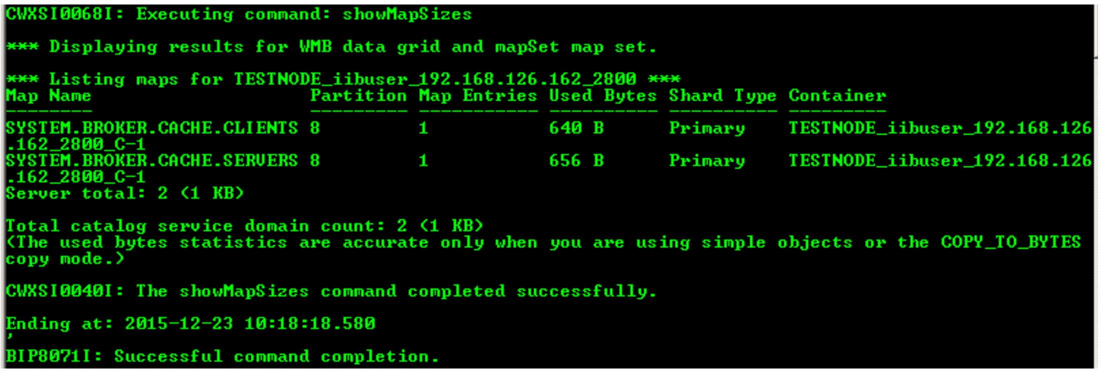
   mqsichangebroker TESTNODE_iibuser -b default

   mqsistart TESTNODE

2. Check the current state of the global caches maps by running the command

   mqsicacheadmin TESTNODE_iibuser -c showMapSizes

This will show output like



```
CWXSI0068I: Executing command: showMapSizes

*** Displaying results for WMB data grid and mapSet map set.

*** Listing maps for TESTNODE_iibuser_192.168.126.162_2800 ***
Map Name                    Partition Map Entries Used Bytes Shard Type Container
--------                    --------- ----------- ---------- ---------- ---------
SYSTEM.BROKER.CACHE.CLIENTS 8         1           640 B      Primary    TESTNODE_iibuser_192.168.126
.162_2800_C-1
SYSTEM.BROKER.CACHE.SERVERS 8         1           656 B      Primary    TESTNODE_iibuser_192.168.126
.162_2800_C-1
Server total: 2 (1 KB)

Total catalog service domain count: 2 (1 KB)
(The used bytes statistics are accurate only when you are using simple objects or the COPY_TO_BYTES
copy mode.)

CWXSI0040I: The showMapSizes command completed successfully.

Ending at: 2015-12-23 10:18:18.580
,
BIP8071I: Successful command completion.
```

Note that the only maps available are the system defaults, each of which currently has 1 entry.

3. Import and test the partial solution

3.1 Import the partial solution

1. To avoid Toolkit workspace conflicts, create a new IIB workspace.

   If you already have a workspace open, click File, Switch Workspace. Give the new workspace the name

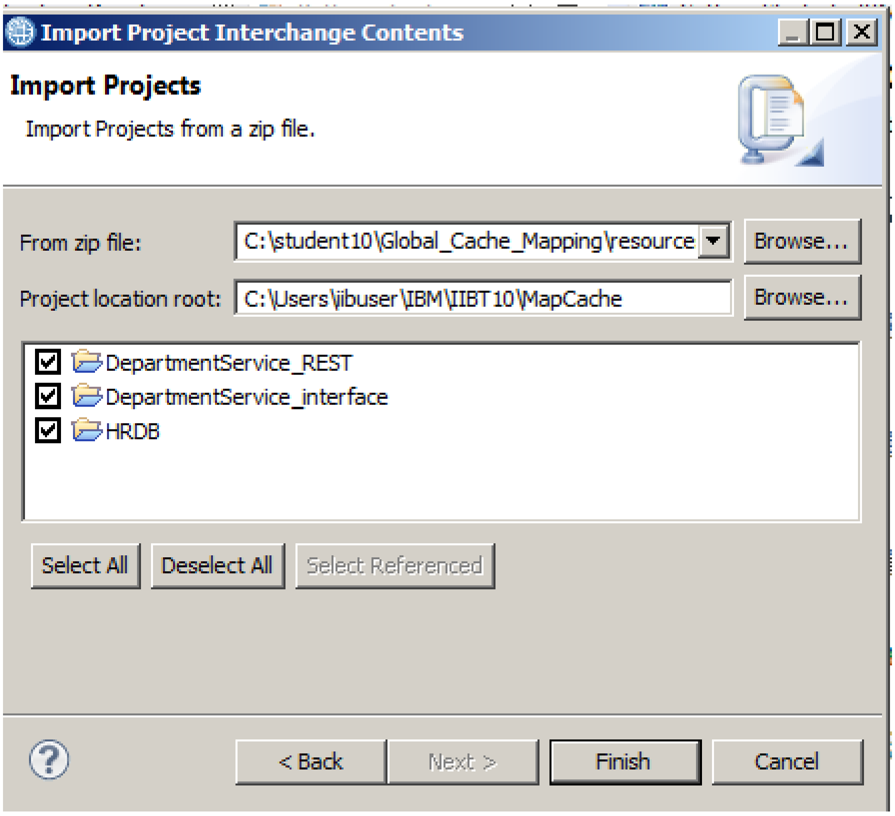
   c:\users\iibuser\IBM\IIB 10\workspace_MapCache

2. In the Application Development window, right-click in an open area (on the white background), and select Import.

3. Select the Project Interchange file

   c:\student10 \ Global_Cache_Mapping \ resources \ DepartmentService_REST.StartingPoint.10.0.0.3.zip

   This PI file contains three projects. Ensure all projects are selected, and click Finish to import them.



Import Project Interchange Contents

Import Projects

Import Projects from a zip file.

From zip file: C:\student10\Global_Cache_Mapping\resource

Project location root: C:\Users\iibuser\IBM\IIBT10\MapCache

☑ DepartmentService_REST
☑ DepartmentService_interface
☑ HRDB

DepartmentService_REST has a library reference to DepartmentService_interface.

DepartmentService_interface is a Shared library that includes the project HRDB.

HRDB is a database definition project that holds the DBM file for the HRDB database.

3.2 Investigate and test the application

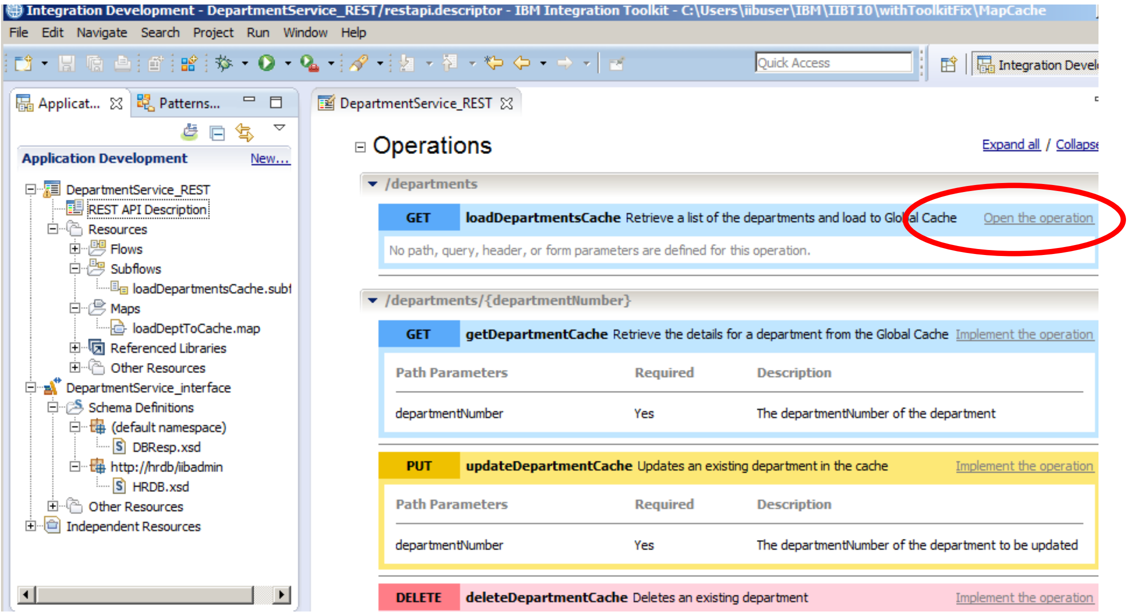
1. Open the DepartmentService_REST application (double-click on the line REST API Description).

   Expand the "twisty" /departments/{departmentNumber}.

   You will see the following REST operations:

   - In the /departments folder:
     - GET - loadDepartmentsToCache
   - In the /departments/{departmentNumber} folder:
     - GET - getDepartmentCache (only partially implemented - you will complete this later)
     - PUT - updateDepartmentCache
     - DELETE - deleteDepartmentCache

   In the /departments section, for the loadDepartmentsCache, click "Open the operation".



Integration Development - DepartmentService_REST/restapi.descriptor - IBM Integration Toolkit - C:\Users\iibuser\IBM\IIBT10\withToolkitFix\MapCache

Operations

/departments

GET **loadDepartmentsCache** Retrieve a list of the departments and load to Global Cache — Open the operation

No path, query, header, or form parameters are defined for this operation.

/departments/{departmentNumber}

GET **getDepartmentCache** Retrieve the details for a department from the Global Cache — Implement the operation

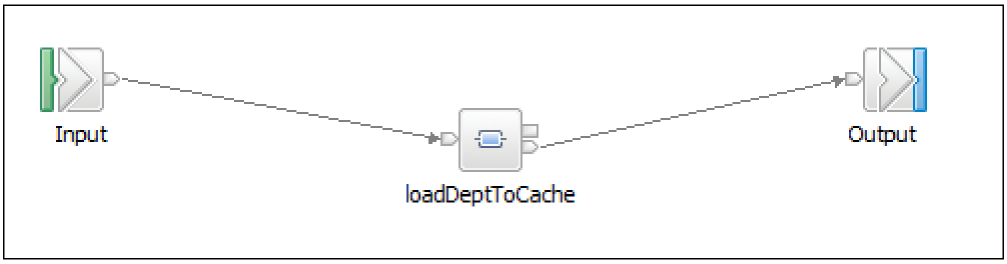
Path Parameters	Required	Description
departmentNumber	Yes	The departmentNumber of the department

PUT **updateDepartmentCache** Updates an existing department in the cache — Implement the operation

Path Parameters	Required	Description
departmentNumber	Yes	The departmentNumber of the department to be updated

DELETE **deleteDepartmentCache** Deletes an existing department — Implement the operation

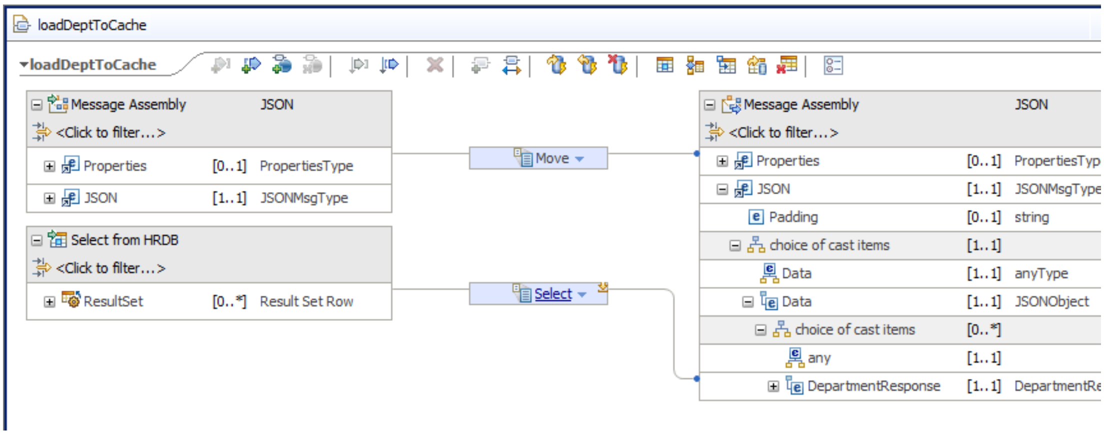
2. The loadDepartmentsCache operation is implemented as a subflow (created automatically when the operation is implemented). This uses a map to retrieve all rows from the DEPARTMENT table, and loads them into the global cache Map1.



3. Open (double-click) the loadDeptToCache map.

   The map retrieves data from the HRDB database, and writes the returned columns to a JSON object cast as the complex type DepartmentResponse.

   We will explore how this is done.



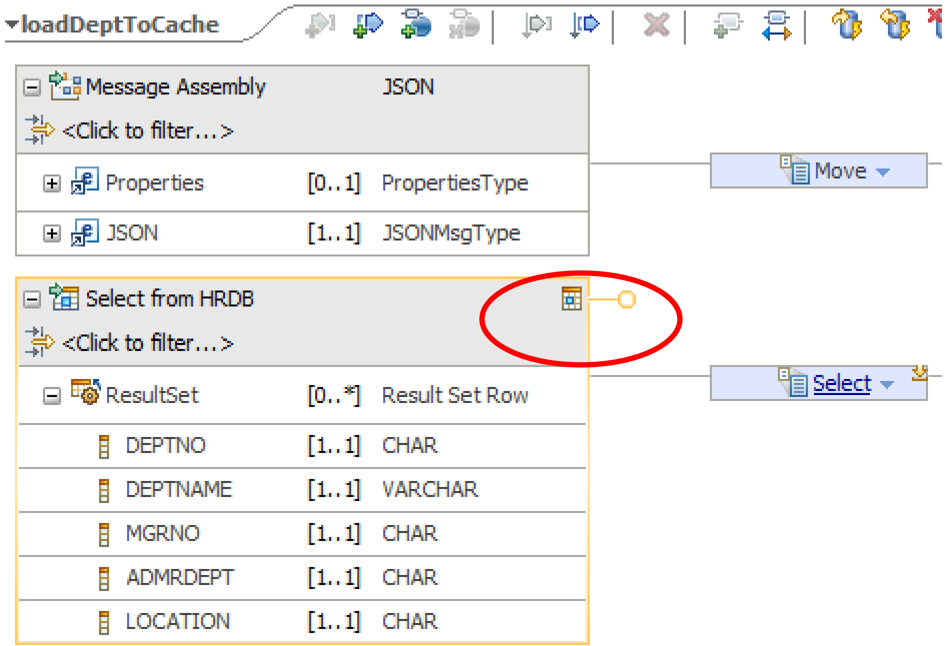
Message Assembly		JSON
<Click to filter...>		
Properties	[0..1]	PropertiesType
JSON	[1..1]	JSONMsgType

Select from HRDB		
<Click to filter...>		
ResultSet	[0..*]	Result Set Row

Message Assembly		JSON
<Click to filter...>		
Properties	[0..1]	PropertiesTyp
JSON	[1..1]	JSONMsgType
Padding	[0..1]	string
choice of cast items	[1..1]	
Data	[1..1]	anyType
Data	[1..1]	JSONObject
choice of cast items	[0..*]	
any	[1..1]	
DepartmentResponse	[1..1]	DepartmentRe

4. First, all rows are retrieved from the DEPARTMENT table in the HRDB database.

   Hover over the "Select from HRDB" input assembly. Click the "read from database" icon (highlighted below).

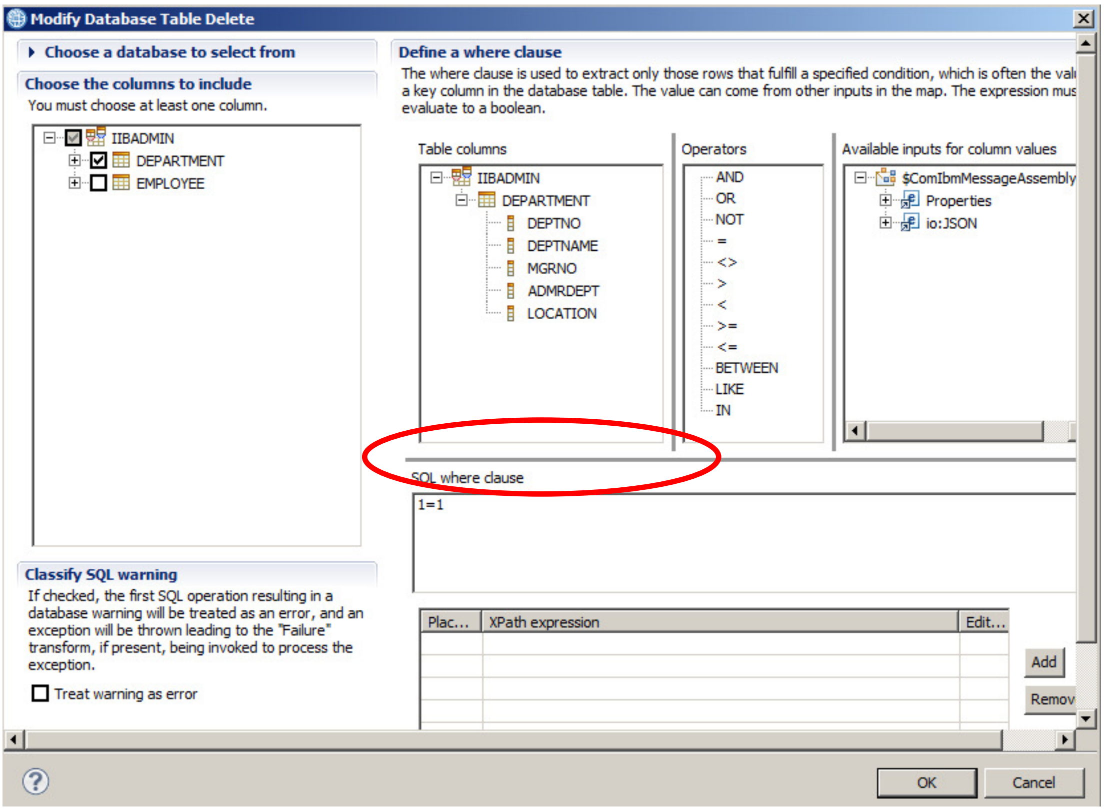


Message Assembly		JSON
<Click to filter...>		
Properties	[0..1]	PropertiesType
JSON	[1..1]	JSONMsgType

Select from HRDB		
<Click to filter...>		
ResultSet	[0..*]	Result Set Row
DEPTNO	[1..1]	CHAR
DEPTNAME	[1..1]	VARCHAR
MGRNO	[1..1]	CHAR
ADMRDEPT	[1..1]	CHAR
LOCATION	[1..1]	CHAR

5. The Database Table wizard will show the format of the SQL statement that is used to retrieve all rows from the DEPARTMENT table. The only selection that was made for this was to select the required tables (DEPARTMENT). The SQL where clause is the default value, and returns all rows.

   When you have reviewed this wizard, click Cancel.



Modify Database Table Delete

Choose a database to select from

Choose the columns to include

You must choose at least one column.

- ☑ IBADMIN
  - ☑ DEPARTMENT
  - □ EMPLOYEE

Classify SQL warning

If checked, the first SQL operation resulting in a database warning will be treated as an error, and an exception will be thrown leading to the "Failure" transform, if present, being invoked to process the exception.

□ Treat warning as error

Define a where clause

The where clause is used to extract only those rows that fulfill a specified condition, which is often the val a key column in the database table. The value can come from other inputs in the map. The expression mus evaluate to a boolean.

Table columns: IIBADMIN > DEPARTMENT > DEPTNO, DEPTNAME, MGRNO, ADMRDEPT, LOCATION

Operators: AND, OR, NOT, =, <>, >, <, >=, <=, BETWEEN, LIKE, IN

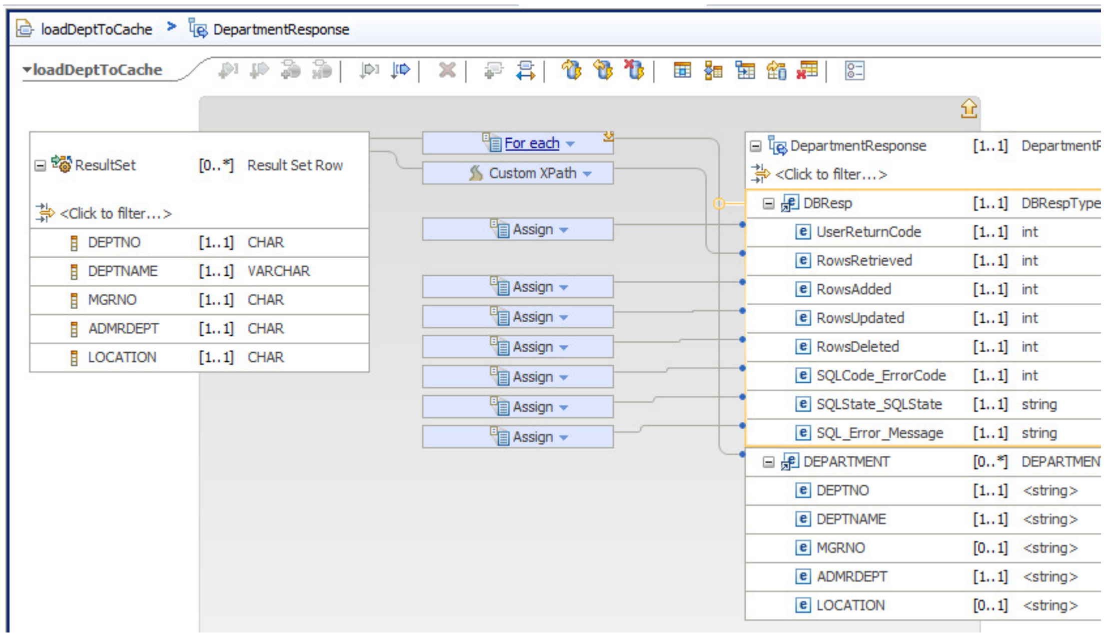
Available inputs for column values: $ComIbmMessageAssembly > Properties, io:JSON

SQL where clause

```
1=1
```

Plac...	XPath expression	Edit...

6. In the map, click "Select" to investigate the details of the transformation. You will see that a "For each" transform has been used to process each row of data returned from the DEPARTMENT table.



loadDeptToCache > DepartmentResponse

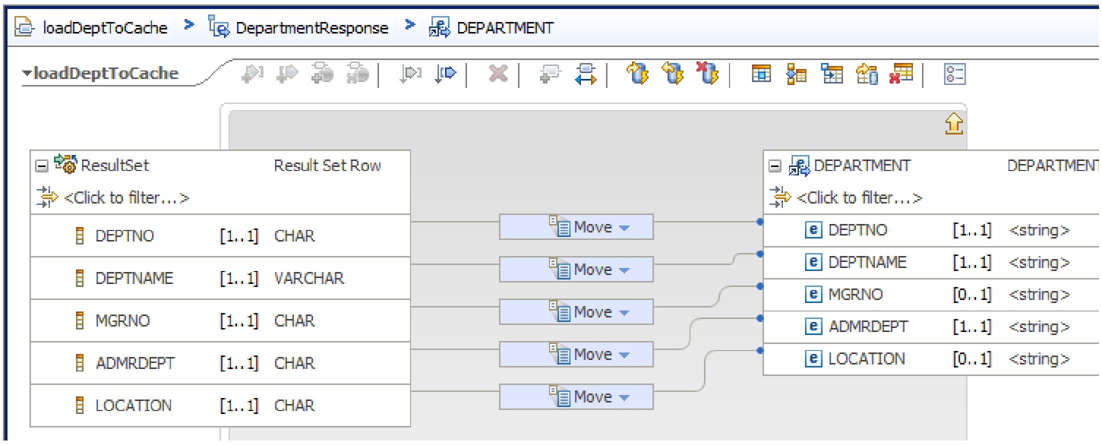
loadDeptToCache

ResultSet	[0..*]	Result Set Row
<Click to filter...>		
DEPTNO	[1..1]	CHAR
DEPTNAME	[1..1]	VARCHAR
MGRNO	[1..1]	CHAR
ADMRDEPT	[1..1]	CHAR
LOCATION	[1..1]	CHAR

DepartmentResponse	[1..1]	DepartmentF
<Click to filter...>		
DBResp	[1..1]	DBRespType
UserReturnCode	[1..1]	int
RowsRetrieved	[1..1]	int
RowsAdded	[1..1]	int
RowsUpdated	[1..1]	int
RowsDeleted	[1..1]	int
SQLCode_ErrorCode	[1..1]	int
SQLState_SQLState	[1..1]	string
SQL_Error_Message	[1..1]	string
DEPARTMENT	[0..*]	DEPARTMEN
DEPTNO	[1..1]	<string>
DEPTNAME	[1..1]	<string>
MGRNO	[0..1]	<string>
ADMRDEPT	[1..1]	<string>
LOCATION	[0..1]	<string>

7. Click "For each" to see the details of how each returned row is handled.

   Each element is connected directly to the output assembly (using Move transforms); this is to show the data after executing the REST application.



loadDeptToCache > DepartmentResponse > DEPARTMENT

loadDeptToCache

ResultSet		Result Set Row
<Click to filter…>		
DEPTNO	[1..1]	CHAR
DEPTNAME	[1..1]	VARCHAR
MGRNO	[1..1]	CHAR
ADMRDEPT	[1..1]	CHAR
LOCATION	[1..1]	CHAR

DEPARTMENT		DEPARTMENT
<Click to filter…>		
DEPTNO	[1..1]	<string>
DEPTNAME	[1..1]	<string>
MGRNO	[0..1]	<string>
ADMRDEPT	[1..1]	<string>
LOCATION	[0..1]	<string>

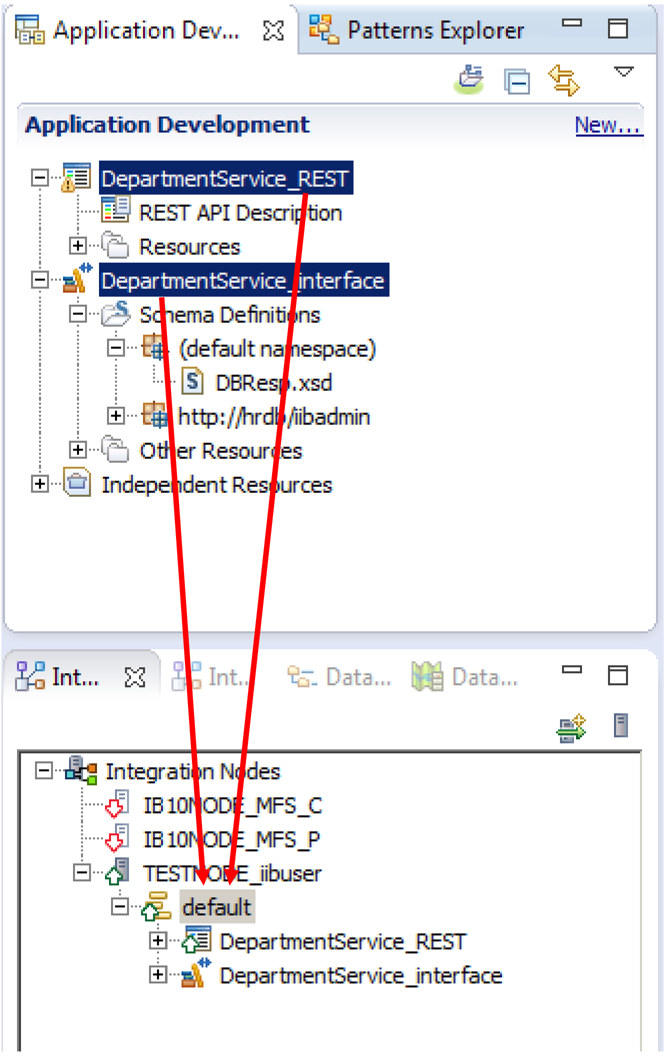
8. Close the loadDeptToCache map now.

   Also close the loadDepartmentsCache subflow.

3.3 Deploy and test the DepartmentService_REST application

1. To make sure the provided REST application runs successfully, deploy the application to the TESTNODE_iibuser/default server.

   The DepartmentService_interface shared library must be deployed first, followed by the DepartmentSservice_REST application.



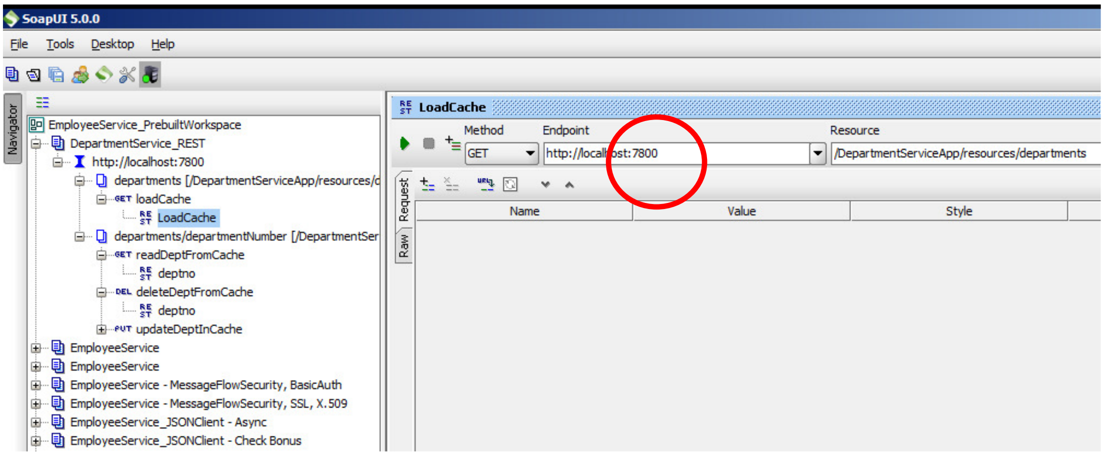
2. Open SOAPUI from the Windows Start menu (or use an existing instance).

   Expand the project `DepartmentService_REST`
```
                   -> http://localhost:7800
                         -> departments
                               -> loadCache
```

   Open the LoadCache request. You will see this has been configured to use a GET method. However, no parameters are provided; the resource is simply set to the URL of the request:

   `/DepartmentServiceApp/resources/departments`

   No input data is required to execute this operation, but make sure the port is set to 7800.

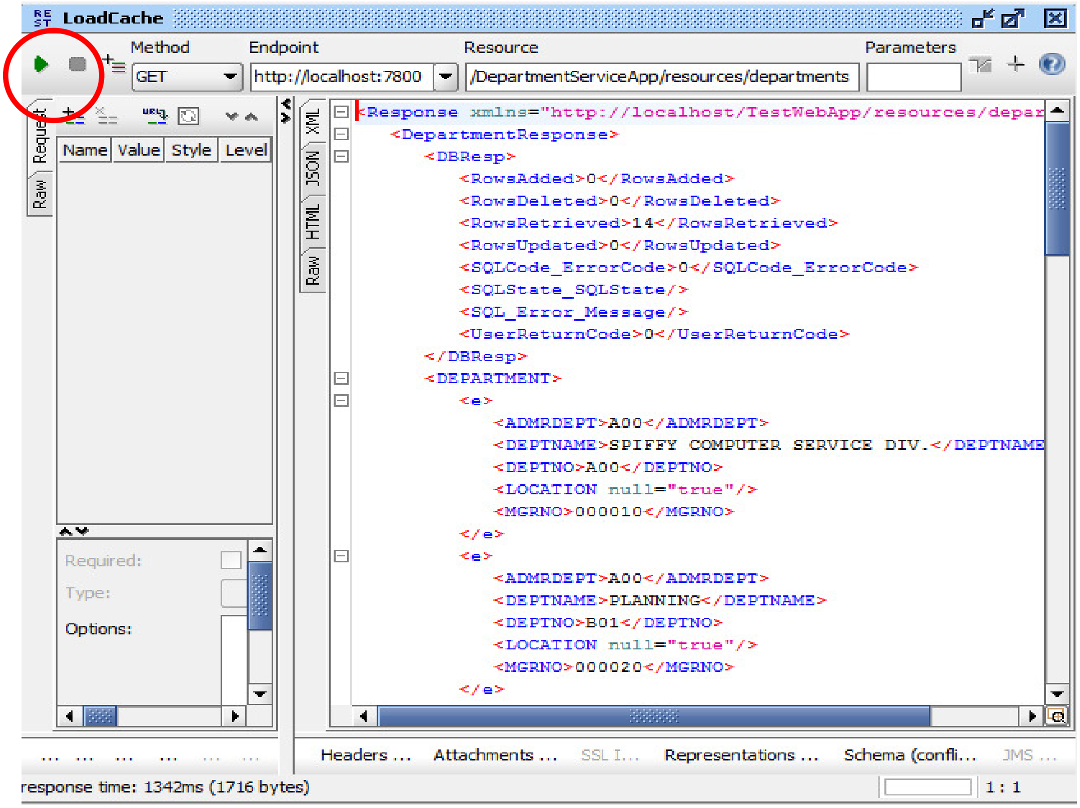

SoapUI 5.0.0
File
Tools
Desktop
Help
Navigator
EmployeeService_PrebuiltWorkspace
DepartmentService_REST
http://localhost:7800
departments [/DepartmentServiceApp/resources/d
GET loadCache
LoadCache
departments/departmentNumber [/DepartmentSer
GET readDeptFromCache
deptno
DEL deleteDeptFromCache
deptno
PUT updateDeptInCache
EmployeeService
EmployeeService
EmployeeService - MessageFlowSecurity, BasicAuth
EmployeeService - MessageFlowSecurity, SSL, X.509
EmployeeService_JSONClient - Async
EmployeeService_JSONClient - Check Bonus
LoadCache
Request
Raw
Name
Value
Style

Method: GET

Endpoint: http://localhost:7800

Resource: /DepartmentServiceApp/resources/departments

3. Click the Green arrow to execute the operation. You will see all rows of the DEPARTMENT table are returned in the JSON array.



Method: GET

Endpoint: http://localhost:7800

Resource: /DepartmentServiceApp/resources/departments

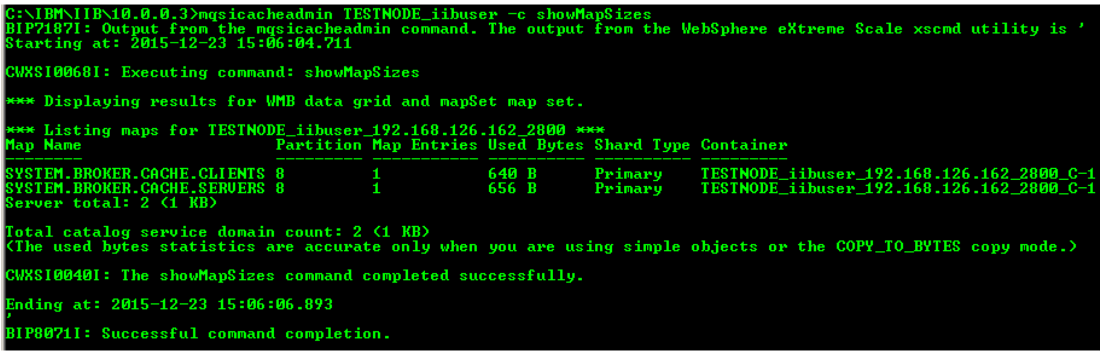
```
<Response xmlns="http://localhost/TestWebApp/resources/depar
   <DepartmentResponse>
      <DBResp>
         <RowsAdded>0</RowsAdded>
         <RowsDeleted>0</RowsDeleted>
         <RowsRetrieved>14</RowsRetrieved>
         <RowsUpdated>0</RowsUpdated>
         <SQLCode_ErrorCode>0</SQLCode_ErrorCode>
         <SQLState_SQLState/>
         <SQL_Error_Message/>
         <UserReturnCode>0</UserReturnCode>
      </DBResp>
      <DEPARTMENT>
         <e>
            <ADMRDEPT>A00</ADMRDEPT>
            <DEPTNAME>SPIFFY COMPUTER SERVICE DIV.</DEPTNAME
            <DEPTNO>A00</DEPTNO>
            <LOCATION null="true"/>
            <MGRNO>000010</MGRNO>
         </e>
         <e>
            <ADMRDEPT>A00</ADMRDEPT>
            <DEPTNAME>PLANNING</DEPTNAME>
            <DEPTNO>B01</DEPTNO>
            <LOCATION null="true"/>
            <MGRNO>000020</MGRNO>
         </e>
```

response time: 1342ms (1716 bytes)

4. In an IIB Command Console, issue the command

```
mqsicacheadmin TESTNODE_iibuser -c showMapSizes
```

The output of this command will be similar to below, and will show that the global cache has no user data entries, and just contains the default cache maps.



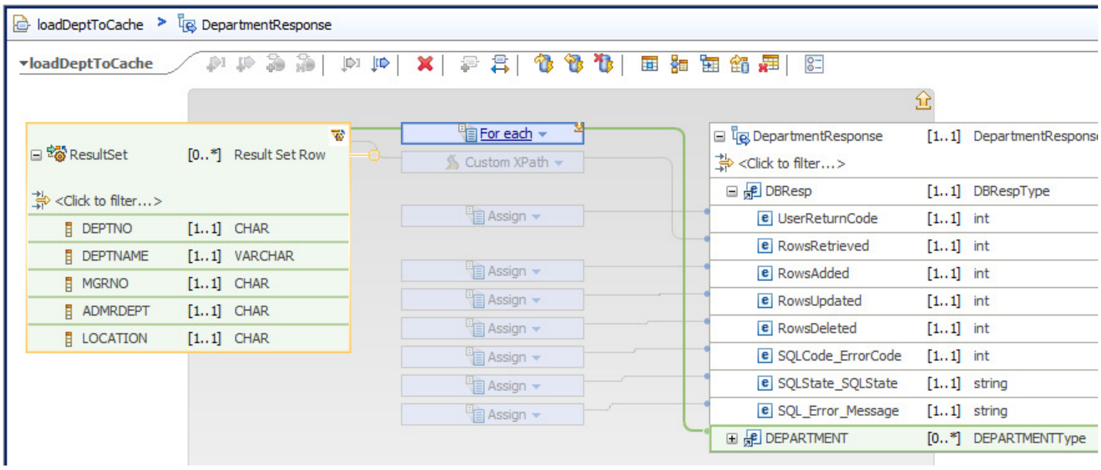
```
C:\IBM\IIB\10.0.0.3>mqsicacheadmin TESTNODE_iibuser -c showMapSizes
BIP7187I: Output from the mqsicacheadmin command. The output from the WebSphere eXtreme Scale xscmd utility is '
Starting at: 2015-12-23 15:06:04.711

CWXSI0068I: Executing command: showMapSizes

*** Displaying results for WMB data grid and mapSet map set.

*** Listing maps for TESTNODE_iibuser_192.168.126.162_2800 ***
Map Name                     Partition Map Entries Used Bytes Shard Type Container
--------                     --------- ----------- ---------- ---------- ---------
SYSTEM.BROKER.CACHE.CLIENTS 8         1           640 B      Primary    TESTNODE_iibuser_192.168.126.162_2800_C-1
SYSTEM.BROKER.CACHE.SERVERS 8         1           656 B      Primary    TESTNODE_iibuser_192.168.126.162_2800_C-1
Server total: 2 (1 KB)

Total catalog service domain count: 2 (1 KB)
(The used bytes statistics are accurate only when you are using simple objects or the COPY_TO_BYTES copy mode.)

CWXSI0040I: The showMapSizes command completed successfully.

Ending at: 2015-12-23 15:06:06.893
'
BIP8071I: Successful command completion.
```

4. Implement the Load Cache operation

You will now extend the loadDepartmentsCache operation to add the rows returned from the DEPARTMENT table into the Global Cache. Earlier, you activated the Global Cache for the TESTNODE_iibuser node; this scenario will use all of the supplied defaults for the global cache, such as cache grid name, map name, and time to live.

1. Open the loadDeptToCache map. and enter the Select transform (click the text "Select").

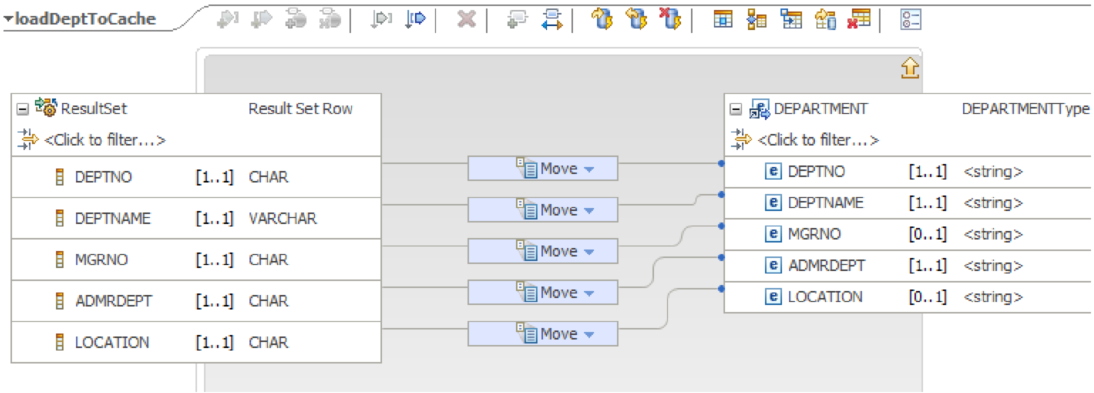
   You will see a number of transforms have been provided. In particular, the "For each" transform takes each row returned from the table, and returns the data to the output JSON message.



ResultSet	[0..*]	Result Set Row
<Click to filter...>		
DEPTNO	[1..1]	CHAR
DEPTNAME	[1..1]	VARCHAR
MGRNO	[1..1]	CHAR
ADMRDEPT	[1..1]	CHAR
LOCATION	[1..1]	CHAR

DepartmentResponse	[1..1]	DepartmentRespons
<Click to filter...>		
DBResp	[1..1]	DBRespType
UserReturnCode	[1..1]	int
RowsRetrieved	[1..1]	int
RowsAdded	[1..1]	int
RowsUpdated	[1..1]	int
RowsDeleted	[1..1]	int
SQLCode_ErrorCode	[1..1]	int
SQLState_SQLState	[1..1]	string
SQL_Error_Message	[1..1]	string
DEPARTMENT	[0..*]	DEPARTMENTType

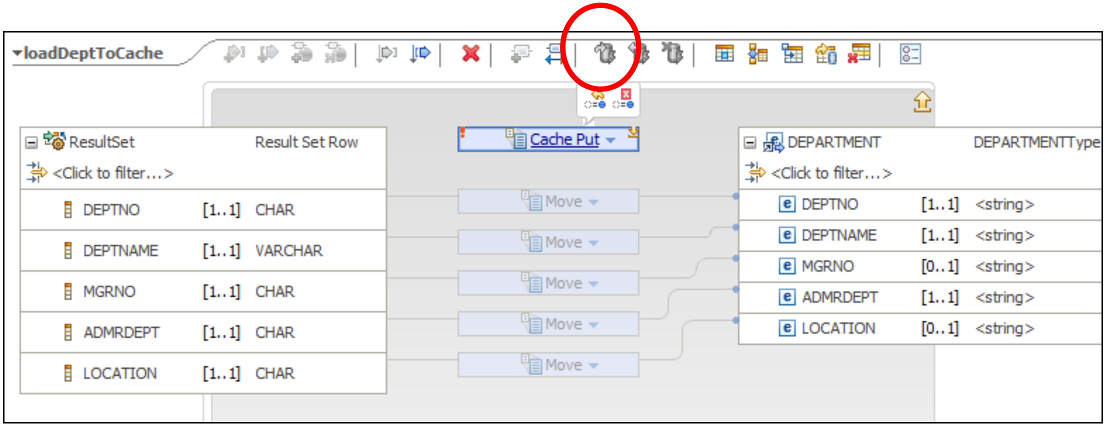
2. Click "For each", which will show the individual element mappings for this.



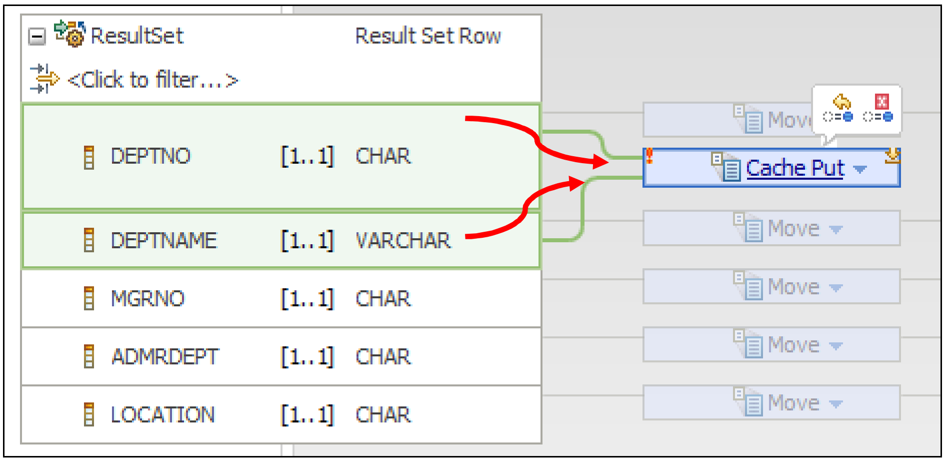
ResultSet		Result Set Row
<Click to filter...>		
DEPTNO	[1..1]	CHAR
DEPTNAME	[1..1]	VARCHAR
MGRNO	[1..1]	CHAR
ADMRDEPT	[1..1]	CHAR
LOCATION	[1..1]	CHAR

DEPARTMENT		DEPARTMENTType
<Click to filter...>		
DEPTNO	[1..1]	<string>
DEPTNAME	[1..1]	<string>
MGRNO	[0..1]	<string>
ADMRDEPT	[1..1]	<string>
LOCATION	[0..1]	<string>

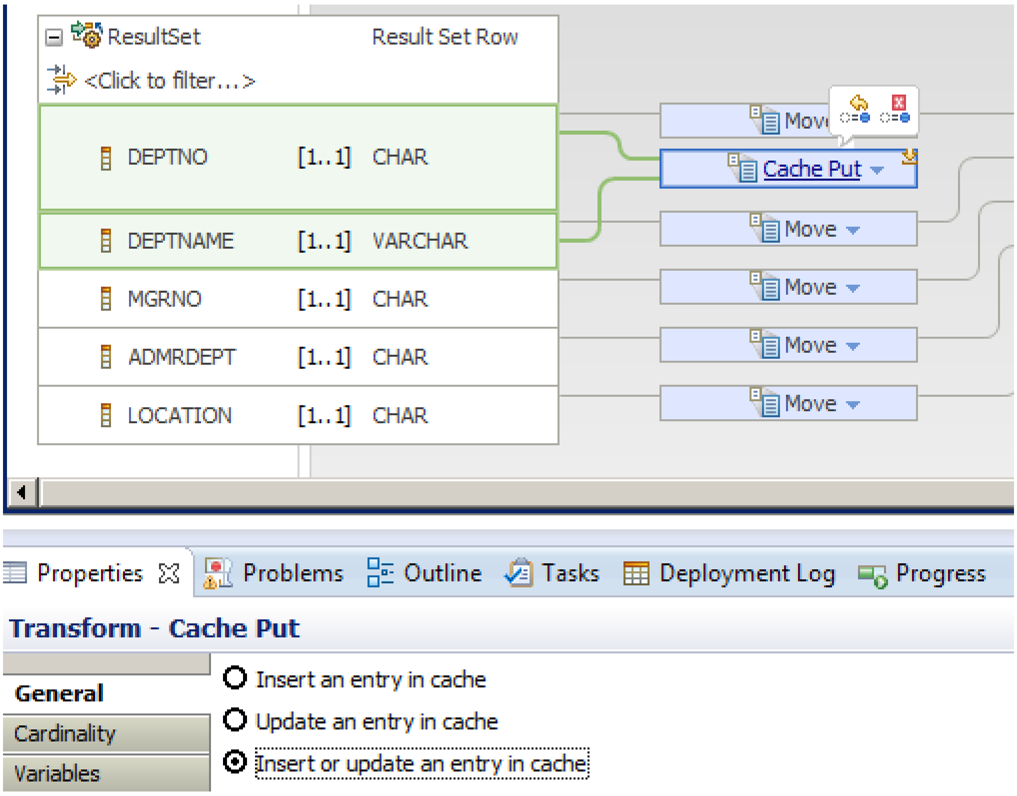
3. Hover over the Cache Put icon (highlighted below), then click the icon to add a Cache Put transform to the map.



4. Connect the input DEPTNO and DEPTNAME elements to the Cache Put transform.



5. Highlight the Cache Put transform, and look at the Properties tab. On the General tab, click the button "Insert or update an entry in the cache". This will mean that existing cache entries will be overwritten if they already exist (which makes testing of this lab easier).



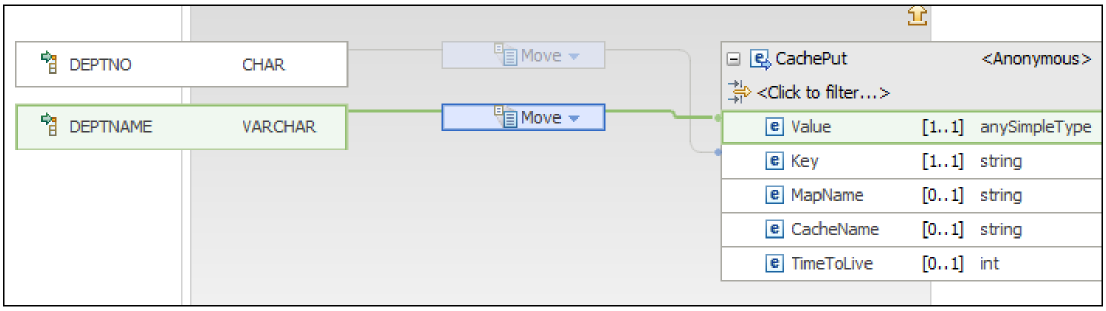
Transform - Cache Put

○ Insert an entry in cache
○ Update an entry in cache
● Insert or update an entry in cache

6. Click the text "Cache Put" to define the individual element mappings.

Connect DEPTNO to Key.

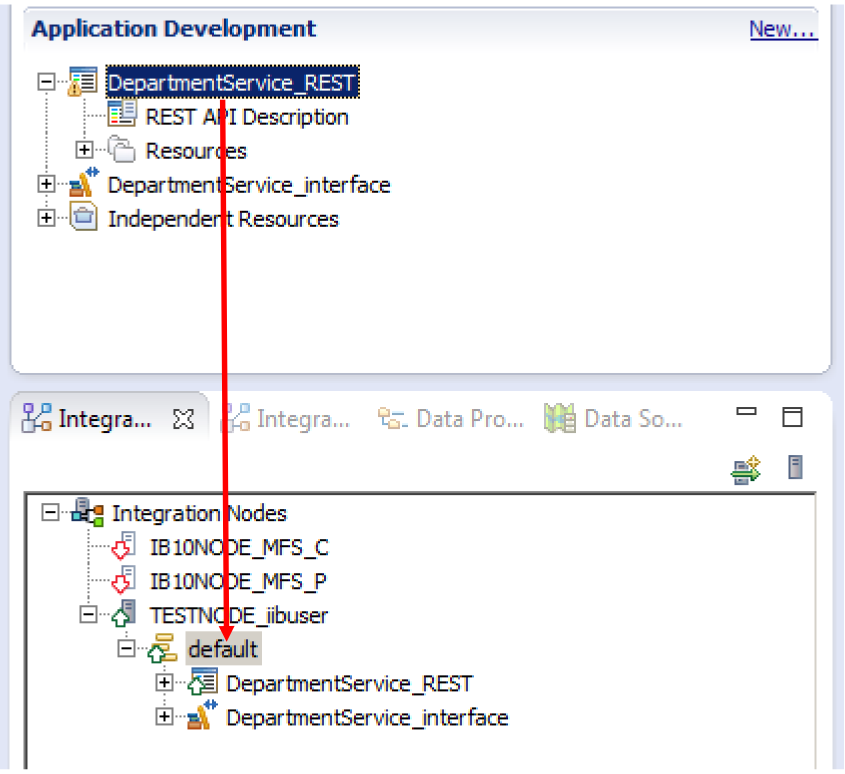
Connect DEPTNAME to Value.



Save the map.

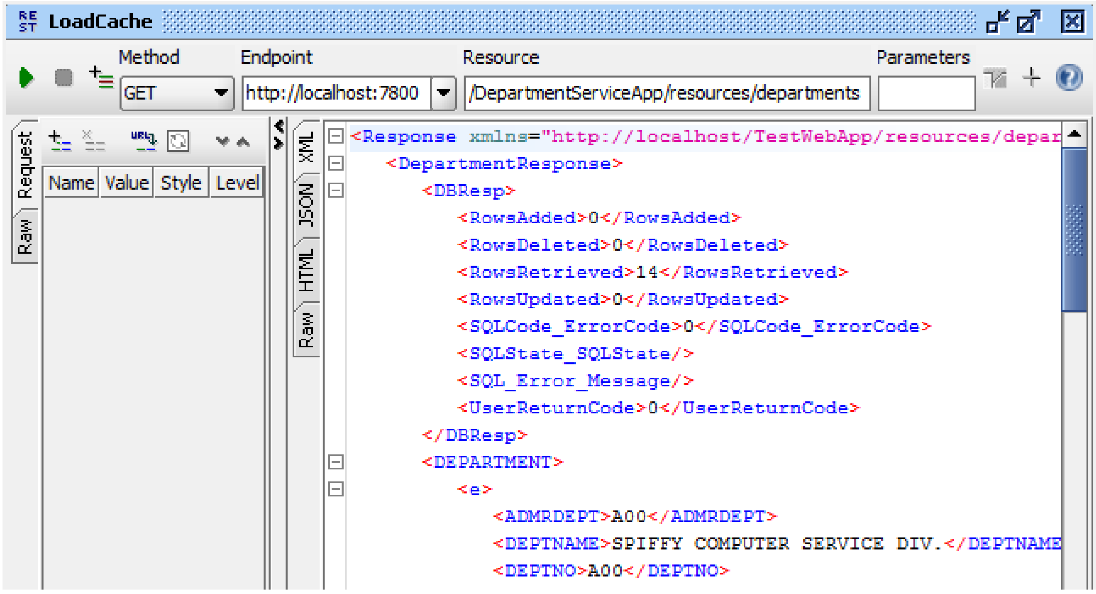
4.1 Redeploy and test the Load cache operation

1. Redeploy the DepartmentService_REST application



2. In SOAPUI, rerun the Load Cache request.

   The same data should be returned from the application.



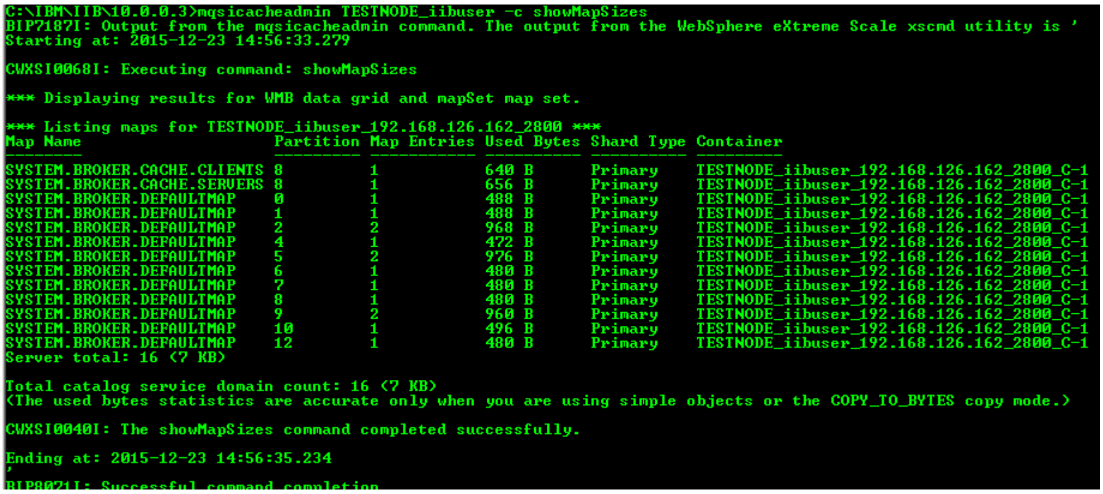
Method: GET
Endpoint: http://localhost:7800
Resource: /DepartmentServiceApp/resources/departments
Parameters:

```
<Response xmlns="http://localhost/TestWebApp/resources/depar
   <DepartmentResponse>
      <DBResp>
         <RowsAdded>0</RowsAdded>
         <RowsDeleted>0</RowsDeleted>
         <RowsRetrieved>14</RowsRetrieved>
         <RowsUpdated>0</RowsUpdated>
         <SQLCode_ErrorCode>0</SQLCode_ErrorCode>
         <SQLState_SQLState/>
         <SQL_Error_Message/>
         <UserReturnCode>0</UserReturnCode>
      </DBResp>
      <DEPARTMENT>
         <e>
            <ADMRDEPT>A00</ADMRDEPT>
            <DEPTNAME>SPIFFY COMPUTER SERVICE DIV.</DEPTNAME
            <DEPTNO>A00</DEPTNO>
```

3. In an IIB Command Console, issue the command

   `mqsicacheadmin TESTNODE_iibuser -c showMapSizes`

   The output of this command will be similar to below, and will show that the global cache has several shards, and that the default cache map has been used. Each shard has either 1 or 2 map entries, representing each database table row.



```
C:\IBM\IIB\10.0.0.3>mqsicacheadmin TESTNODE_iibuser -c showMapSizes
BIP7187I: Output from the mqsicacheadmin command. The output from the WebSphere eXtreme Scale xscmd utility is '
Starting at: 2015-12-23 14:56:33.279

CWXSI0068I: Executing command: showMapSizes

*** Displaying results for WMB data grid and mapSet map set.

*** Listing maps for TESTNODE_iibuser_192.168.126.162_2800 ***
Map Name                    Partition Map Entries Used Bytes Shard Type Container
--------                    --------- ----------- ---------- ---------- ---------
SYSTEM.BROKER.CACHE.CLIENTS 8         1           640 B      Primary    TESTNODE_iibuser_192.168.126.162_2800_C-1
SYSTEM.BROKER.CACHE.SERVERS 8         1           656 B      Primary    TESTNODE_iibuser_192.168.126.162_2800_C-1
SYSTEM.BROKER.DEFAULTMAP    0         1           488 B      Primary    TESTNODE_iibuser_192.168.126.162_2800_C-1
SYSTEM.BROKER.DEFAULTMAP    1         1           488 B      Primary    TESTNODE_iibuser_192.168.126.162_2800_C-1
SYSTEM.BROKER.DEFAULTMAP    2         2           968 B      Primary    TESTNODE_iibuser_192.168.126.162_2800_C-1
SYSTEM.BROKER.DEFAULTMAP    4         1           472 B      Primary    TESTNODE_iibuser_192.168.126.162_2800_C-1
SYSTEM.BROKER.DEFAULTMAP    5         2           976 B      Primary    TESTNODE_iibuser_192.168.126.162_2800_C-1
SYSTEM.BROKER.DEFAULTMAP    6         1           480 B      Primary    TESTNODE_iibuser_192.168.126.162_2800_C-1
SYSTEM.BROKER.DEFAULTMAP    7         1           480 B      Primary    TESTNODE_iibuser_192.168.126.162_2800_C-1
SYSTEM.BROKER.DEFAULTMAP    8         1           480 B      Primary    TESTNODE_iibuser_192.168.126.162_2800_C-1
SYSTEM.BROKER.DEFAULTMAP    9         2           960 B      Primary    TESTNODE_iibuser_192.168.126.162_2800_C-1
SYSTEM.BROKER.DEFAULTMAP    10        1           496 B      Primary    TESTNODE_iibuser_192.168.126.162_2800_C-1
SYSTEM.BROKER.DEFAULTMAP    12        1           480 B      Primary    TESTNODE_iibuser_192.168.126.162_2800_C-1
Server total: 16 (7 KB)

Total catalog service domain count: 16 (7 KB)
(The used bytes statistics are accurate only when you are using simple objects or the COPY_TO_BYTES copy mode.)

CWXSI0040I: The showMapSizes command completed successfully.

Ending at: 2015-12-23 14:56:35.234
'
BIP8071I: Successful command completion
```

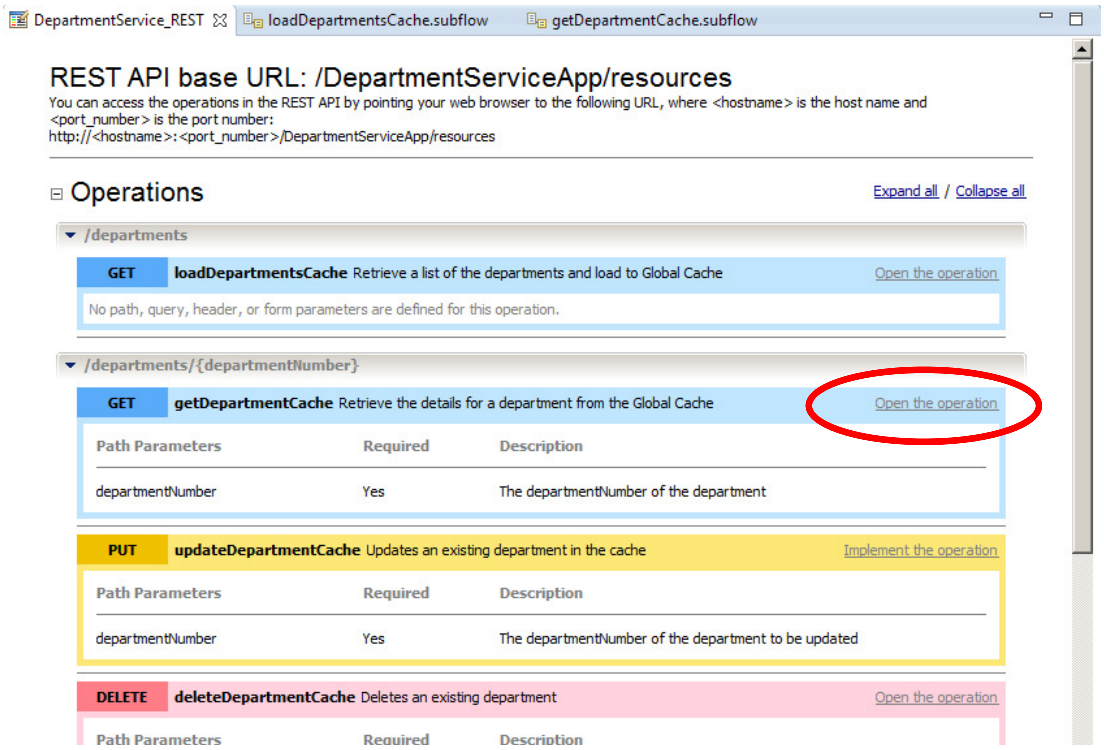
You can rerun the loadCache operation, since the Cache Put transform has been set to update or insert.

5. Implement the Get Department Cache operation

1. You will now implement the getDepartmentCache operation.

   In the Operations list, locate getDepartmentCache and click "Open the operation".

   This will open the subflow editor. This operation has been created, but its implementation is incomplete.


DepartmentService_REST
loadDepartmentsCache.subflow
getDepartmentCache.subflow

REST API base URL: /DepartmentServiceApp/resources

You can access the operations in the REST API by pointing your web browser to the following URL, where <hostname> is the host name and <port_number> is the port number:
http://<hostname>:<port_number>/DepartmentServiceApp/resources

Operations

Expand all / Collapse all

/departments

GET **loadDepartmentsCache** Retrieve a list of the departments and load to Global Cache — Open the operation

No path, query, header, or form parameters are defined for this operation.

/departments/{departmentNumber}

GET **getDepartmentCache** Retrieve the details for a department from the Global Cache — Open the operation

Path Parameters	Required	Description
departmentNumber	Yes	The departmentNumber of the department

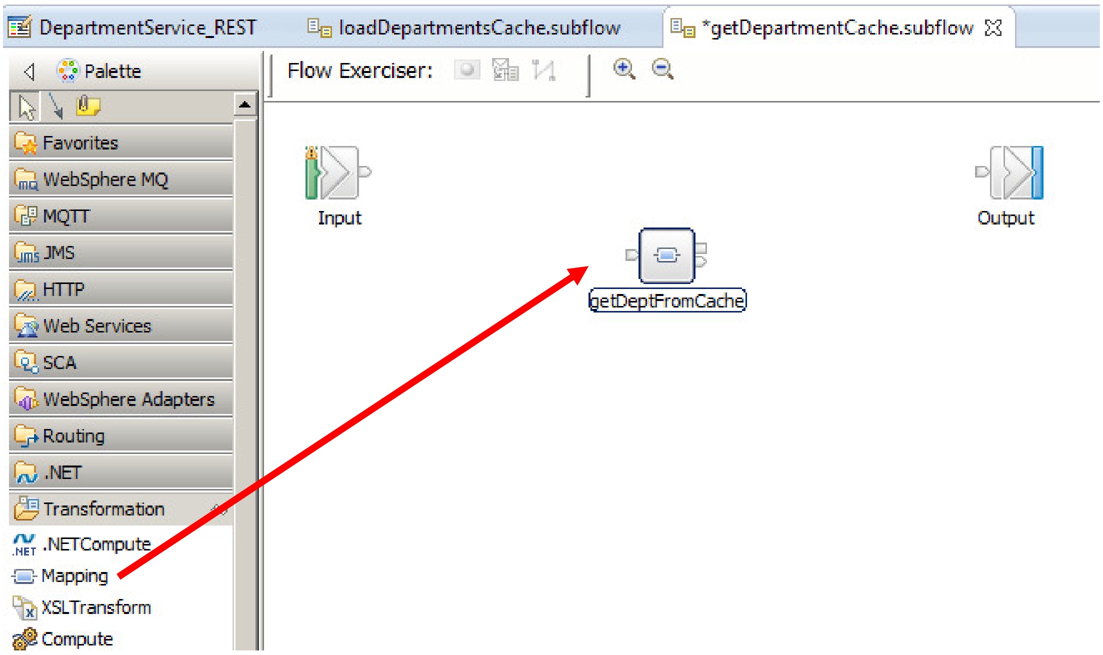
PUT **updateDepartmentCache** Updates an existing department in the cache — Implement the operation

Path Parameters	Required	Description
departmentNumber	Yes	The departmentNumber of the department to be updated

DELETE **deleteDepartmentCache** Deletes an existing department — Open the operation

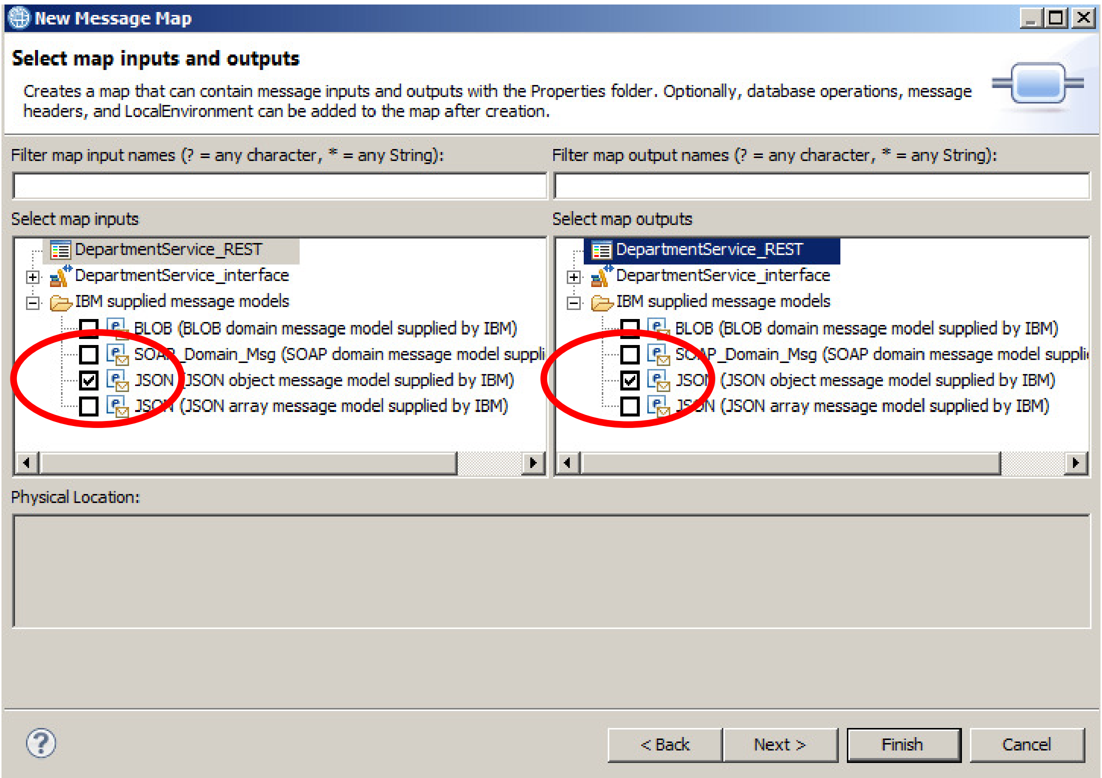
Path Parameters	Required	Description

2. Drop a Mapping node onto the flow editor. Name the new map "getDeptFromCache.



3. Open (double-click) the map, and click Next at the first window.

   For the map inputs and outputs, select JSON object for the input, and JSON Object for the output. Click Finish.



New Message Map

Select map inputs and outputs

Creates a map that can contain message inputs and outputs with the Properties folder. Optionally, database operations, message headers, and LocalEnvironment can be added to the map after creation.

Filter map input names (? = any character, * = any String):

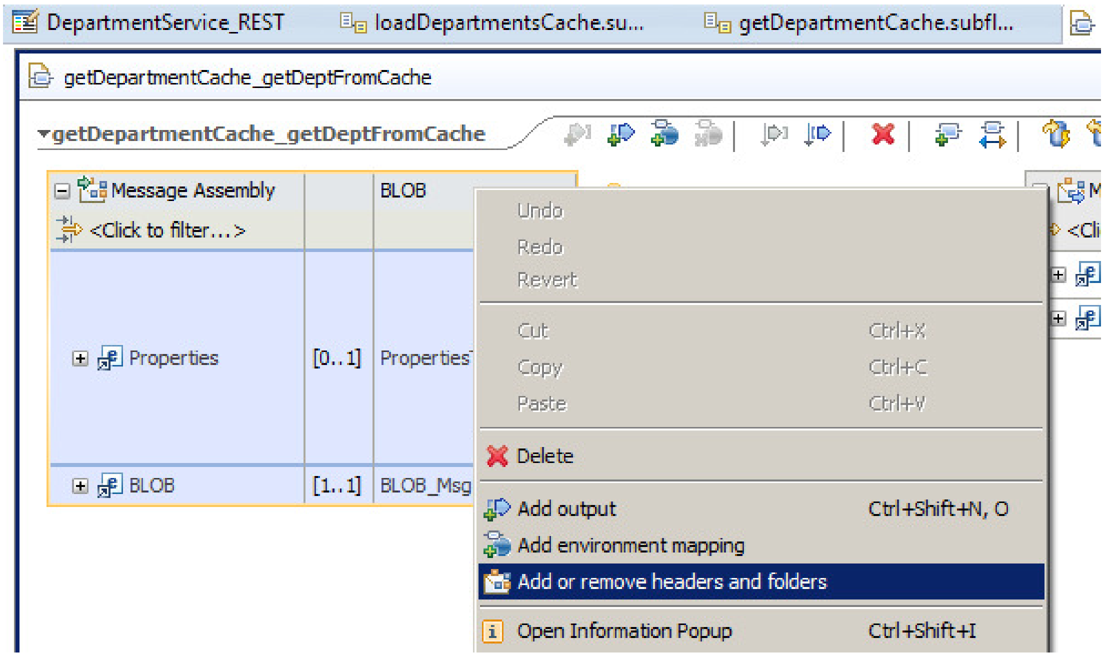
Filter map output names (? = any character, * = any String):

Select map inputs
- DepartmentService_REST
- DepartmentService_interface
- IBM supplied message models
  - □ BLOB (BLOB domain message model supplied by IBM)
  - □ SOAP_Domain_Msg (SOAP domain message model suppli
  - ☑ JSON (JSON object message model supplied by IBM)
  - □ JSON (JSON array message model supplied by IBM)

Select map outputs
- DepartmentService_REST
- DepartmentService_interface
- IBM supplied message models
  - □ BLOB (BLOB domain message model supplied by IBM)
  - □ SOAP_Domain_Msg (SOAP domain message model suppli
  - ☑ JSON (JSON object message model supplied by IBM)
  - □ JSON (JSON array message model supplied by IBM)

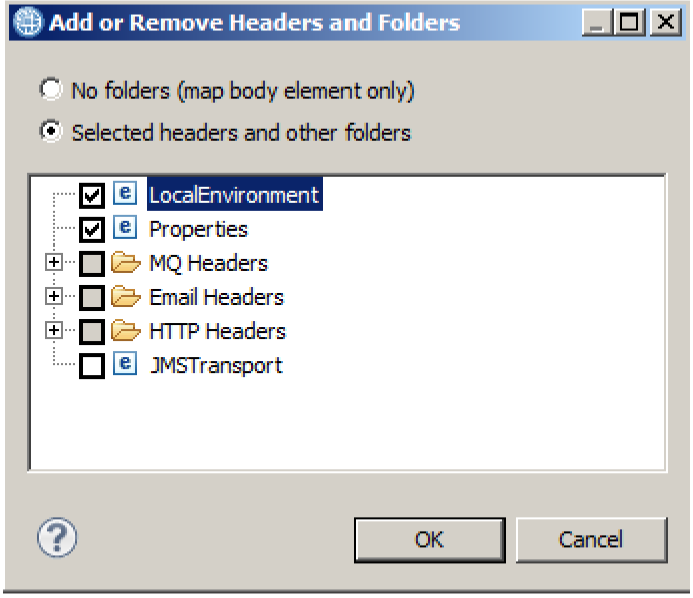
Physical Location:

4. Add the Local Environment to the input assembly. Do this by right-clicking the input assembly (BLOB) and select "Add or remove headers.....".



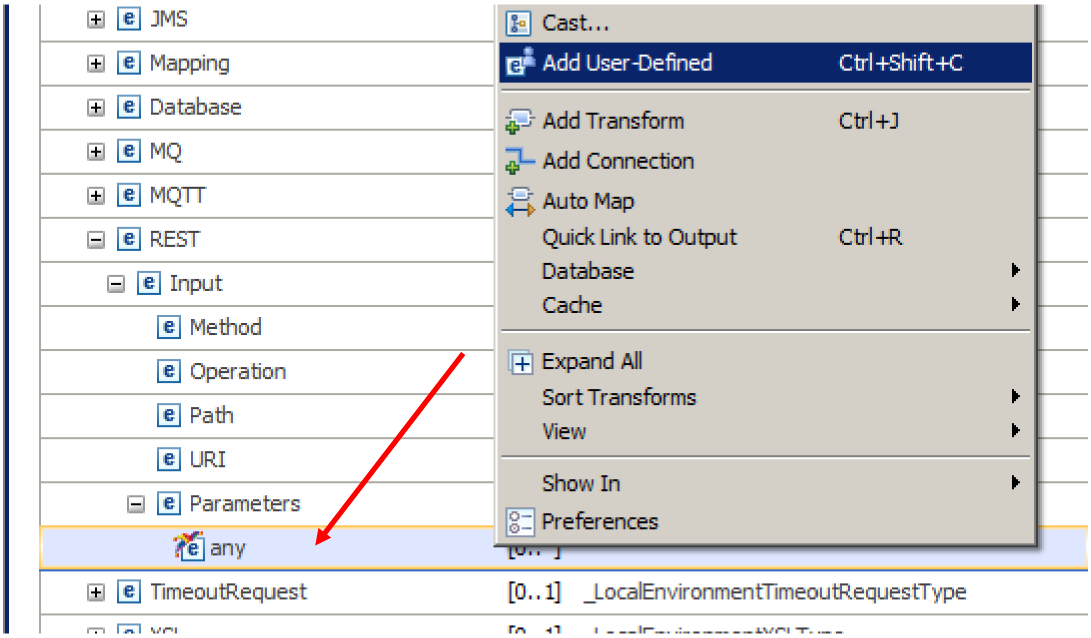
DepartmentService_REST
loadDepartmentsCache.su...
getDepartmentCache.subfl...
getDepartmentCache_getDeptFromCache
getDepartmentCache_getDeptFromCache
Message Assembly
BLOB
<Click to filter...>
Properties
[0..1]
Properties
BLOB
[1..1]
BLOB_Msg
Undo
Redo
Revert
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Delete
Add output Ctrl+Shift+N, O
Add environment mapping
Add or remove headers and folders
Open Information Popup Ctrl+Shift+I

5. Select LocalEnvironment and click OK.

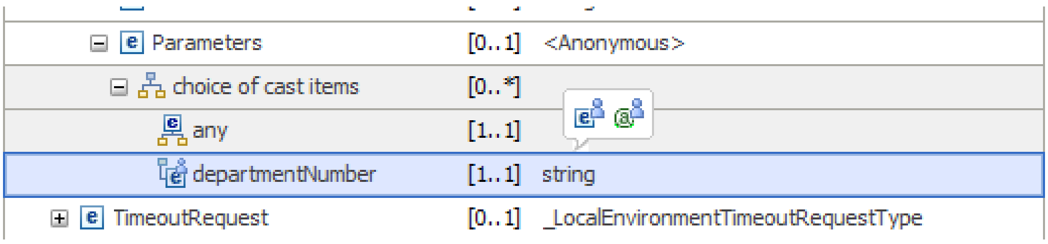

Add or Remove Headers and Folders
○ No folders (map body element only)
● Selected headers and other folders
☑ LocalEnvironment
☑ Properties
□ MQ Headers
□ Email Headers
□ HTTP Headers
□ JMSTransport
OK
Cancel

6. Expand the input assembly Local Environment, and fully expand the REST folder.

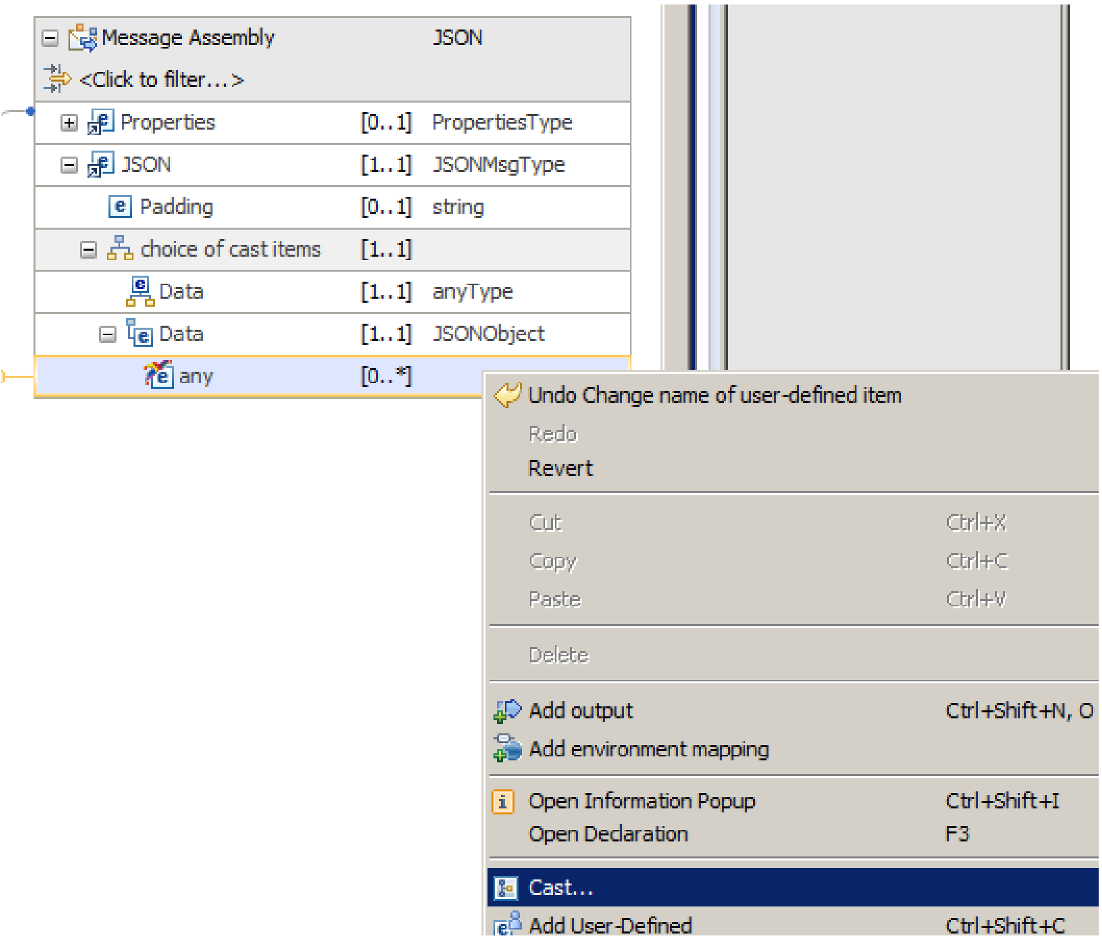
Right-click the "any" element, and select "Add User-defined".



Set the name of the new element to "**departmentNumber**".


Parameters [0..1] <Anonymous>
choice of cast items [0..*]
any [1..1]
departmentNumber [1..1] string
TimeoutRequest [0..1] _LocalEnvironmentTimeoutRequestType

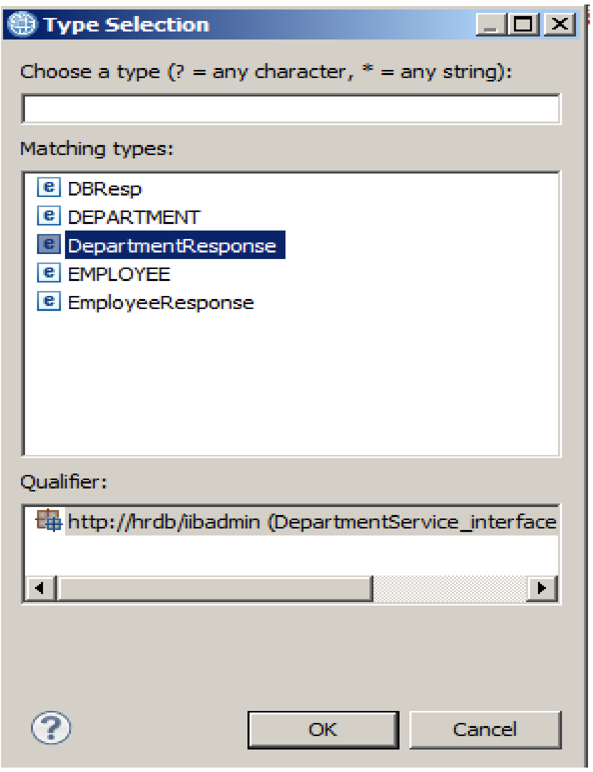
7. In the output message assembly, fully expand the JSON message. Under the "any" element, Right-click and select Cast.



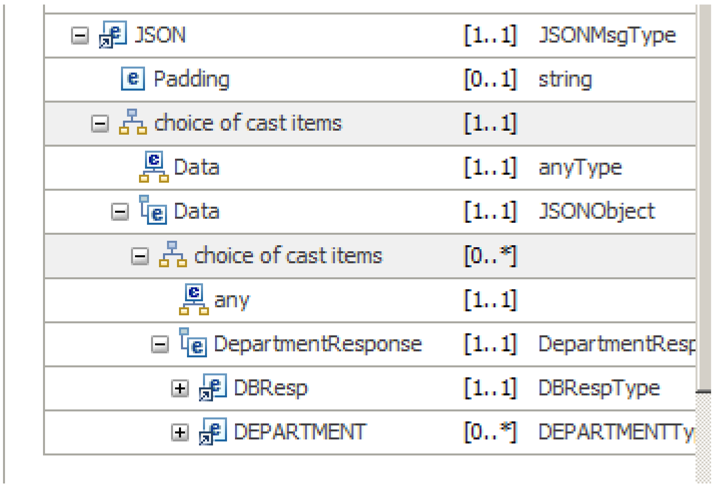
Message Assembly		JSON
<Click to filter...>		
Properties	[0..1]	PropertiesType
JSON	[1..1]	JSONMsgType
Padding	[0..1]	string
choice of cast items	[1..1]	
Data	[1..1]	anyType
Data	[1..1]	JSONObject
any	[0..*]	

Undo Change name of user-defined item
Redo
Revert
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Delete
Add output Ctrl+Shift+N, O
Add environment mapping
Open Information Popup Ctrl+Shift+I
Open Declaration F3
Cast...
Add User-Defined Ctrl+Shift+C

8. In the Type Selection, select DepartmentResponse, and click OK.

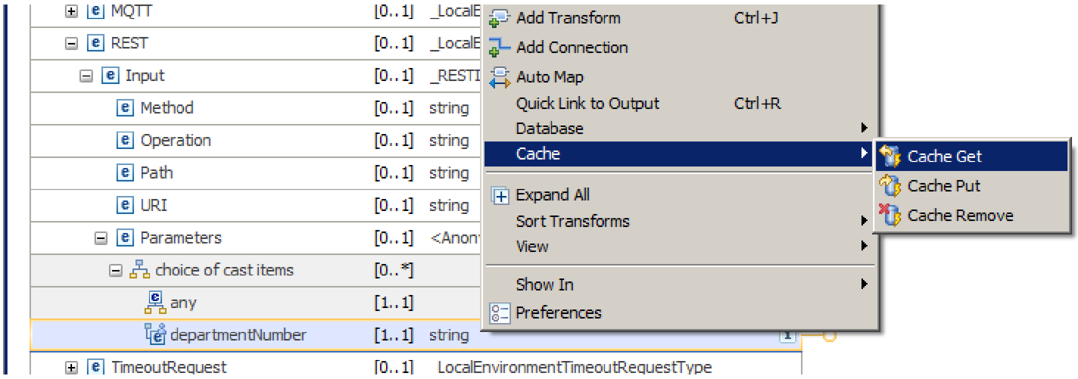


9. The output JSON assembly should now look like this:



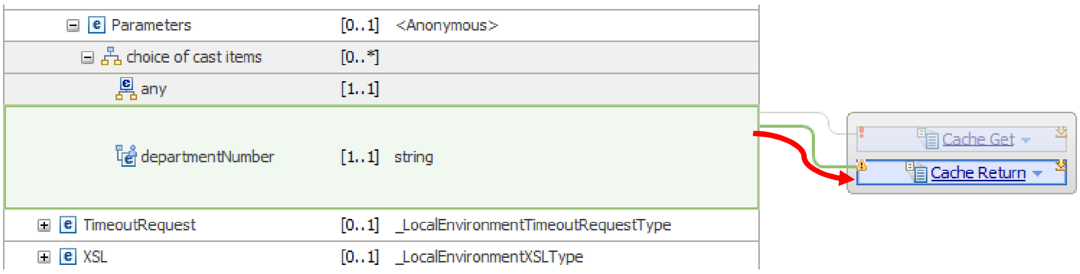
JSON	[1..1]	JSONMsgType
Padding	[0..1]	string
choice of cast items	[1..1]	
Data	[1..1]	anyType
Data	[1..1]	JSONObject
choice of cast items	[0..*]	
any	[1..1]	
DepartmentResponse	[1..1]	DepartmentResp
DBResp	[1..1]	DBRespType
DEPARTMENT	[0..*]	DEPARTMENTTy

10. Back in the input assembly, right-click the new departmentNumber element, and select Cache -> Cache Get.
This will automatically create a Cache Get transform with the appropriate input connection.



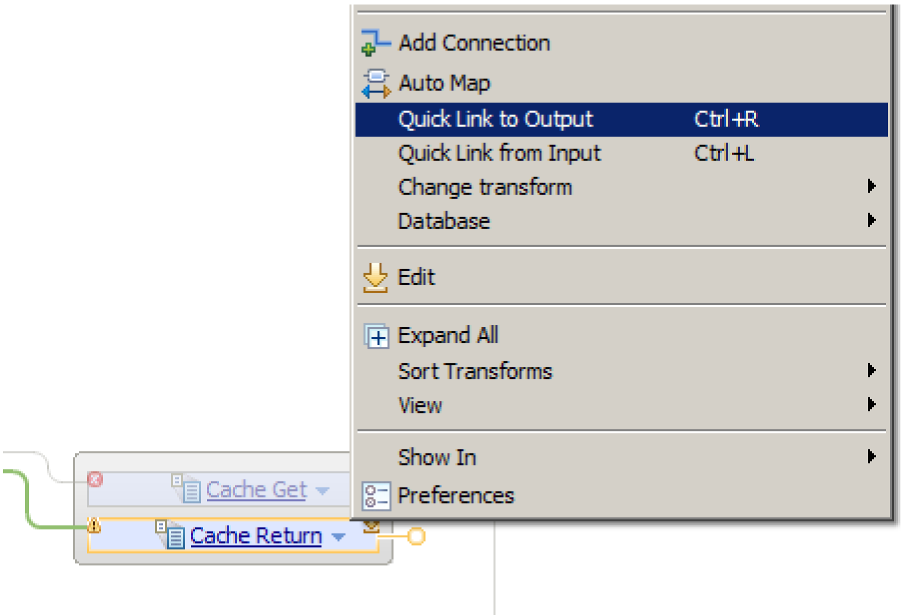
MQTT	[0..1]	_LocalE
REST	[0..1]	_LocalE
Input	[0..1]	_RESTI
Method	[0..1]	string
Operation	[0..1]	string
Path	[0..1]	string
URI	[0..1]	string
Parameters	[0..1]	<Anon
choice of cast items	[0..*]	
any	[1..1]	
departmentNumber	[1..1]	string
TimeoutRequest	[0..1]	LocalEnvironmentTimeoutRequestType

11. We want the input data (departmentNumber) to be returned to the invoking client, so also connect the department Number element to the Cache Return transform.

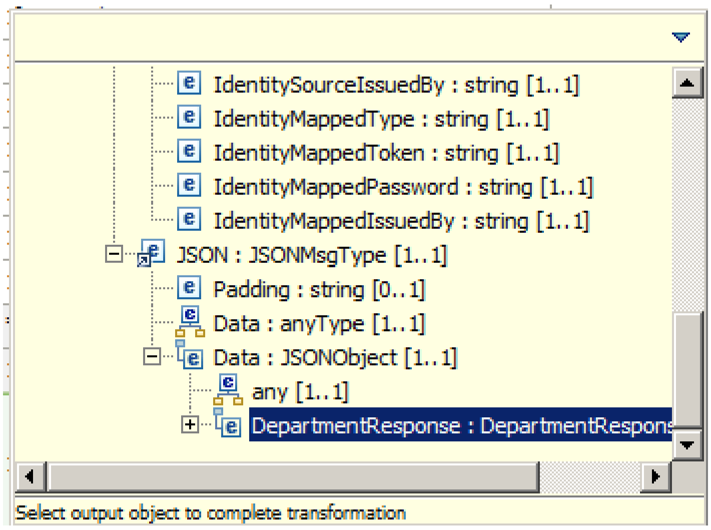


Parameters	[0..1]	<Anonymous>
choice of cast items	[0..*]	
any	[1..1]	
departmentNumber	[1..1]	string
TimeoutRequest	[0..1]	_LocalEnvironmentTimeoutRequestType
XSL	[0..1]	_LocalEnvironmentXSLType

12. We want the data held in the cache to be returned to the invoking client system, so connect the Cache Return to the JSON output. Do this by right-clicking the Cache Return transform, and select "Quick Link to Output".

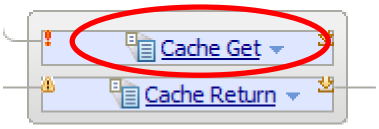


In the pop-up window, near the bottom of the window, expand JSON, Data, and click "DepartmentResponse", as shown below.

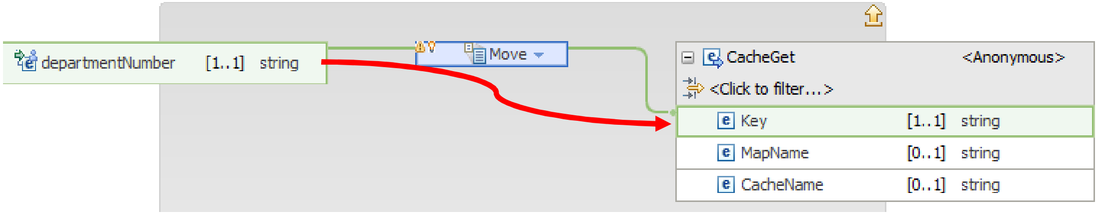

IdentitySourceIssuedBy : string [1..1]
IdentityMappedType : string [1..1]
IdentityMappedToken : string [1..1]
IdentityMappedPassword : string [1..1]
IdentityMappedIssuedBy : string [1..1]
JSON : JSONMsgType [1..1]
Padding : string [0..1]
Data : anyType [1..1]
Data : JSONObject [1..1]
any [1..1]
DepartmentResponse : DepartmentRespons
Select output object to complete transformation

13. You will now navigate to the next level of the Cache Get and Return transforms to specify the individual elements mappings for the transforms.

    First, click the Cache Get transform (click the words "Cache Get").



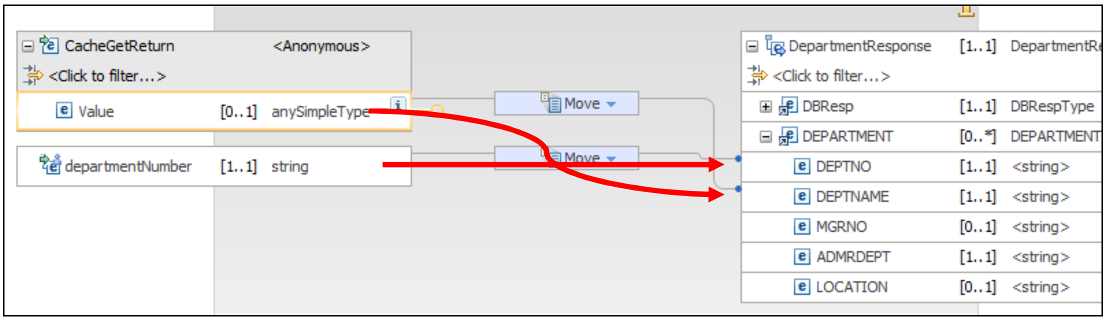
    In the new map level, connect departmentNumber to Key. This will generate a Move transform.



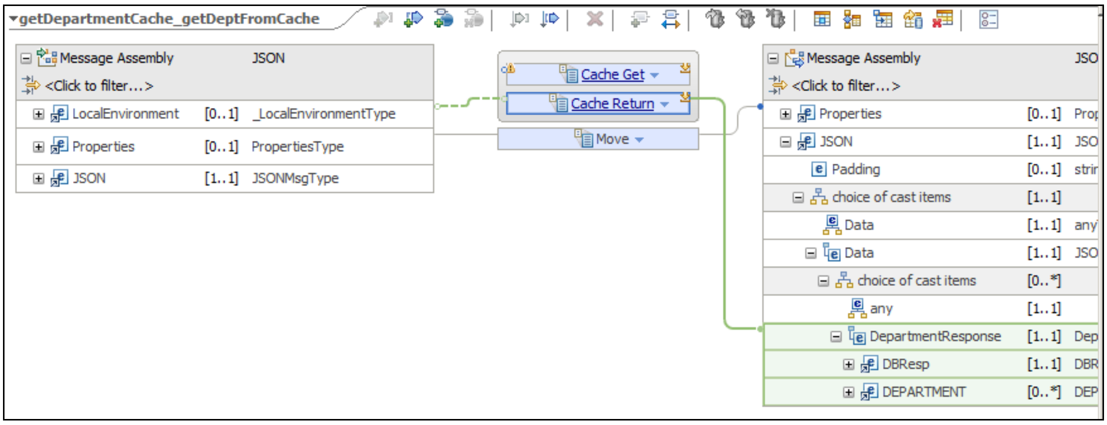
14. Now implement the Cache Return transform. In the map, go up one logical level (yellow arrow), and then click Cache Return (the text "Cache Return").

    This will open the map editor for the Cache Return. Connect Value to deptname and connect departmentNumber to deptno. This will also generate Move transform.

    Note that departNumber appears in the input assembly because you connected the departmentNumber input element to the Cache Return transform.



15. Go up one level in the map. When complete, the map will look like this.



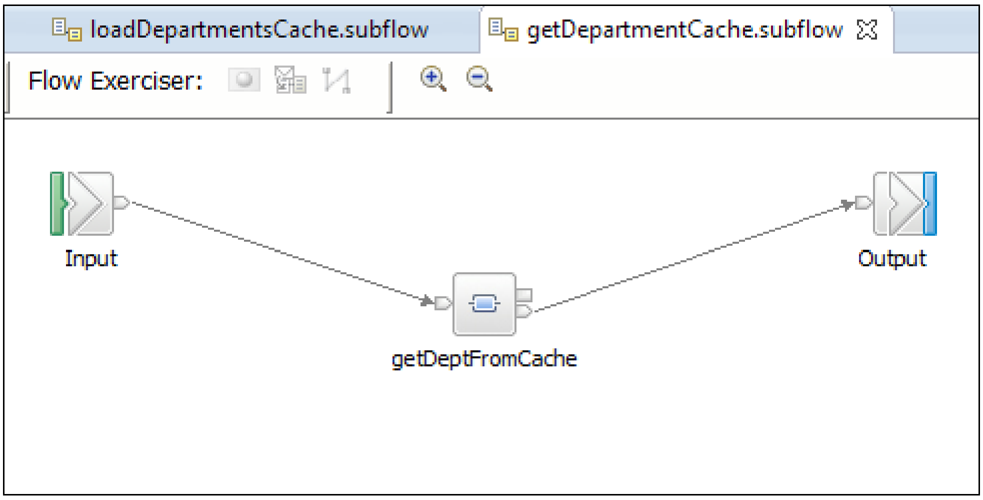
getDepartmentCache_getDeptFromCache

Message Assembly		JSON
<Click to filter...>		
LocalEnvironment	[0..1]	_LocalEnvironmentType
Properties	[0..1]	PropertiesType
JSON	[1..1]	JSONMsgType

Message Assembly		JSO
<Click to filter...>		
Properties	[0..1]	Prop
JSON	[1..1]	JSO
Padding	[0..1]	strir
choice of cast items	[1..1]	
Data	[1..1]	any
Data	[1..1]	JSO
choice of cast items	[0..*]	
any	[1..1]	
DepartmentResponse	[1..1]	Dep
DBResp	[1..1]	DBR
DEPARTMENT	[0..*]	DEP

Save and close the map.

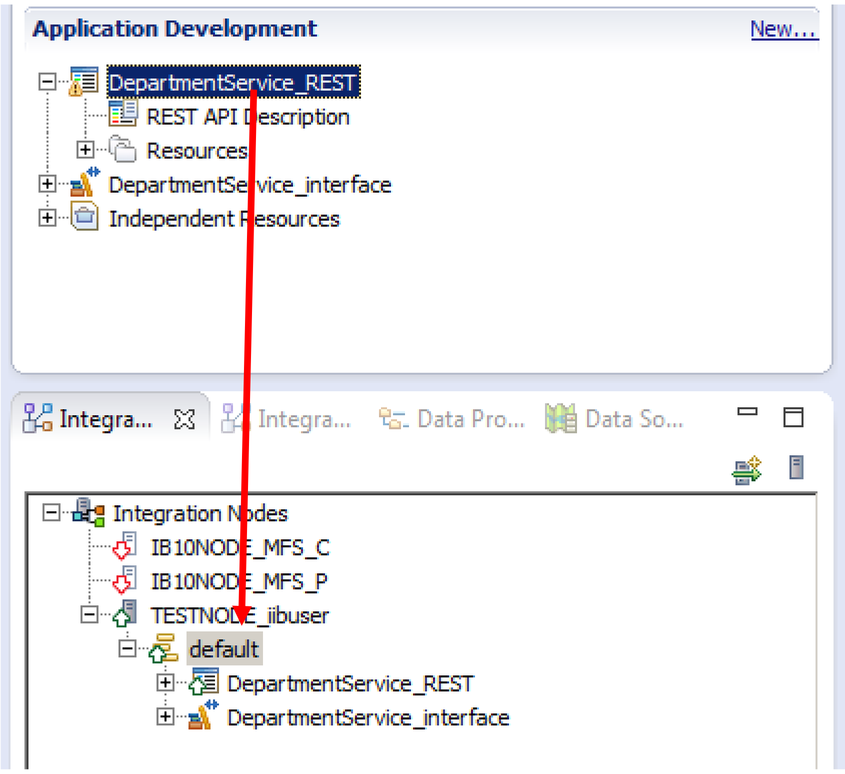
16. In the subflow, connect the nodes as shown, save and close.



5.1 Redeploy and test the getDeptFromCache operation

In addition to using SOAPUI, this chapter will show you how to use the SwaggerUI tool to send a REST request into the REST API service that you have just created.

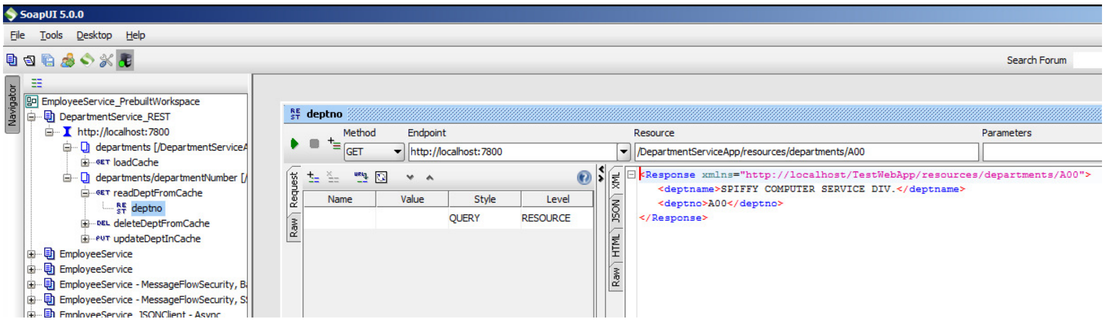
1. In the navigator, redeploy the DepartmentService_REST application to TESTNODE_iibuser/default.



2. In SOAPUI, open the DepartmentService_REST project, and fully expand the readDeptFromCache operation.

   Open the deptno method. Observe that this operation will send the department "A00" for lookup in the cache.

   Click the green arrow to invoke the operation. The data will be returned from the cache.



Method: GET
Endpoint: http://localhost:7800
Resource: /DepartmentServiceApp/resources/departments/A00
Parameters:

Name	Value	Style	Level
		QUERY	RESOURCE

```
<Response xmlns="http://localhost/TestWebApp/resources/departments/A00">
   <deptname>SPIFFY COMPUTER SERVICE DIV.</deptname>
   <deptno>A00</deptno>
</Response>
```

Try other departments such as C01, D01 (change the final value on the URL resource).

3. Alternatively, you can use SwaggerUI to test the REST application.

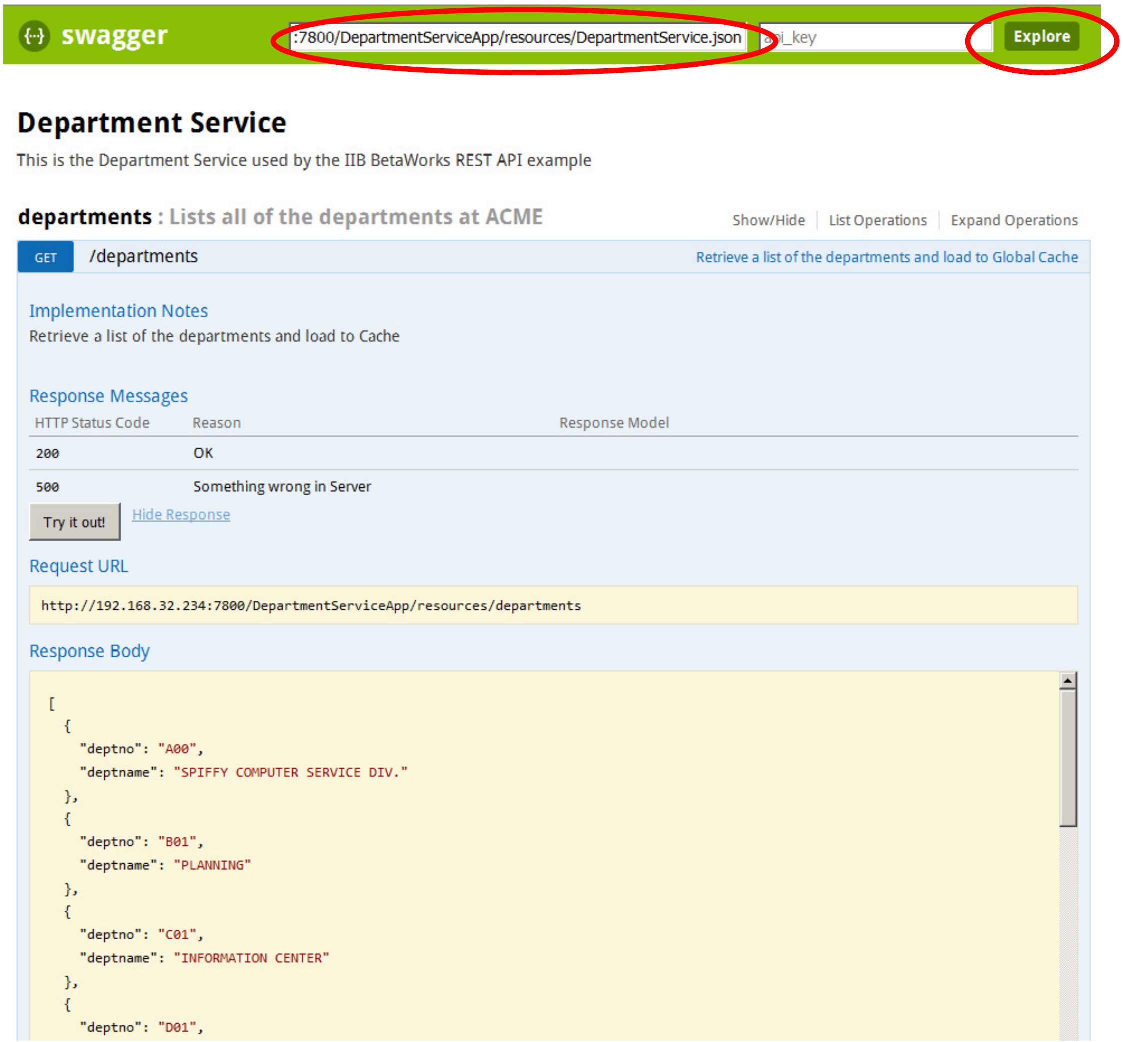
   If you are not familiar with SwaggerUI, review the EmployeeService_REST lab.

   Log in to the IIB Admin browser for RESTNODE, using the bookmark in the REST folder of the Firefox browser.

   Expand the DepartmentService_REST application, and from the API folder, copy the URL for remote JSON invocation (right-click, copy Link Location).

   In a new browser tab, open SwaggerUI and paste the clipboard contents into the input field. Click Explore

   Use SwaggerUI to load the global cache. Run the GET departments operation, which will return all departments, and load them to the cache (click Try it out!).



Department Service

This is the Department Service used by the IIB BetaWorks REST API example

departments : Lists all of the departments at ACME

/departments

Retrieve a list of the departments and load to Global Cache

Implementation Notes

Retrieve a list of the departments and load to Cache

Response Messages

HTTP Status Code	Reason	Response Model
200	OK	
500	Something wrong in Server	

Request URL

```
http://192.168.32.234:7800/DepartmentServiceApp/resources/departments
```

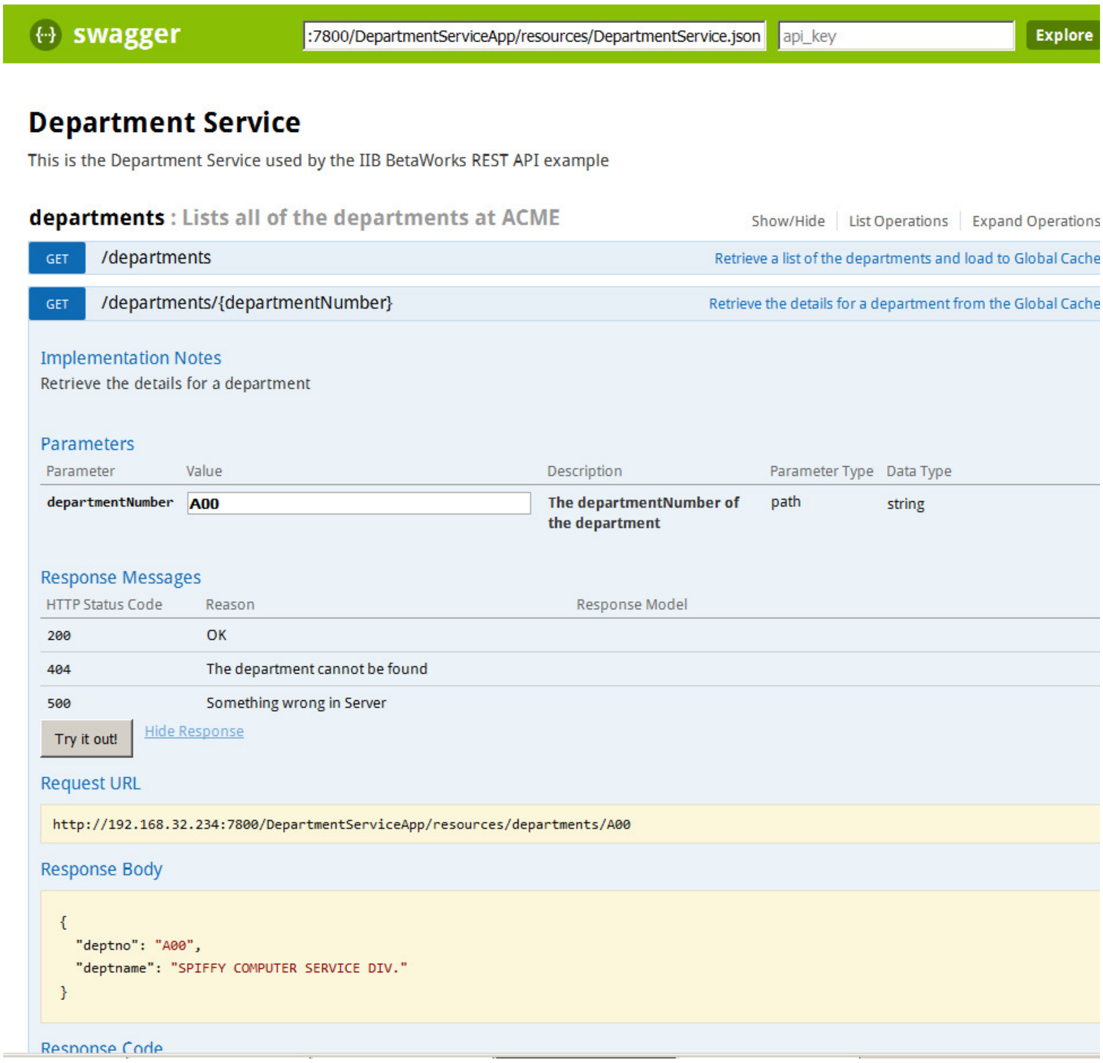
Response Body

```
[
  {
    "deptno": "A00",
    "deptname": "SPIFFY COMPUTER SERVICE DIV."
  },
  {
    "deptno": "B01",
    "deptname": "PLANNING"
  },
  {
    "deptno": "C01",
    "deptname": "INFORMATION CENTER"
  },
  {
    "deptno": "D01",
```

4. In SwaggerUI, expand the departments/departNumber operation.

   Provide a department number, say A00, and click "Try me out!".

   The data will be returned from the cache.



Department Service

This is the Department Service used by the IIB BetaWorks REST API example

departments : Lists all of the departments at ACME

GET /departments — Retrieve a list of the departments and load to Global Cache

GET /departments/{departmentNumber} — Retrieve the details for a department from the Global Cache

Implementation Notes

Retrieve the details for a department

Parameters

Parameter	Value	Description	Parameter Type	Data Type
departmentNumber	A00	The departmentNumber of the department	path	string

Response Messages

HTTP Status Code	Reason	Response Model
200	OK	
404	The department cannot be found	
500	Something wrong in Server	

Request URL

```
http://192.168.32.234:7800/DepartmentServiceApp/resources/departments/A00
```

Response Body

```
{
  "deptno": "A00",
  "deptname": "SPIFFY COMPUTER SERVICE DIV."
}
```

Response Code

END OF LAB GUIDE