

WebSphere Lab Jam

Connectivity

WebSphere Cast Iron

Lab Exercises



Catalog Number

Contents

WEBSHERE CAST IRON - CONNECTION PARAMETERS SPREADSHEET	5
LAB 1	A CAST IRON HELLO..... 6
1.1	OVERVIEW 6
1.2	CONCEPTS..... 6
1.3	ACTIVITIES USED 6
1.4	DOCUMENT KEYS 6
1.5	CAST IRON HELLO 7
1.6	ENHANCED CAST IRON HELLO 19
LAB 2	SINGLE TABLE DATABASE..... 29
2.1	OVERVIEW 29
2.2	CONCEPTS..... 29
2.3	ACTIVITIES USED 29
2.4	DOCUMENT KEYS 29
2.5	INSERT SINGLE ACCOUNT INTO DATABASE 30
2.6	RETURN ALL ACCOUNTS FROM DATABASE 48
APPENDIX A.	NOTICES 55
APPENDIX B.	TRADEMARKS AND COPYRIGHTS 57

THIS PAGE INTENTIONALLY LEFT BLANK

Connection Parameters Spreadsheet

Student Suffix	DB2 Database Schema	WMC ID	WMC Password
ID's for Morning Lab Session 1			
01	TEAM01	Team01	Passw0rd01
02	TEAM02	Team02	Passw0rd02
03	TEAM03	Team03	Passw0rd03
04	TEAM04	Team04	Passw0rd04
05	TEAM05	Team05	Passw0rd05
06	TEAM06	Team06	Passw0rd06
07	TEAM07	Team07	Passw0rd07
08	TEAM08	Team08	Passw0rd08
09	TEAM09	Team09	Passw0rd09
10	TEAM10	Team10	Passw0rd10
ID's for Afternoon Lab Session 2			
11	TEAM11	Team11	Passw0rd11
12	TEAM12	Team12	Passw0rd12
13	TEAM13	Team13	Passw0rd13
14	TEAM14	Team14	Passw0rd14
15	TEAM15	Team15	Passw0rd15
16	TEAM16	Team16	Passw0rd16
17	TEAM17	Team17	Passw0rd17
18	TEAM18	Team18	Passw0rd18
19	TEAM19	Team19	Passw0rd19
20	TEAM20	Team20	Passw0rd20
Endpoint Server Address (DB2):		XX.XX.XX.XX	
Appliance EMgmt Address (WMC):		XX.XX.XX.XX	
Appliance EData Address (URL):		XX.XX.XX.XX	
Database Name		CASTIRON	
Database Table Name		ACCOUNT	
Database User		db2admin	
Database Password		db2admin	
Salesforce User			
Salesforce password + security token			
Path to Lab Resources		C:\Student\Lab Resources	

Lab 1 A Cast Iron Hello

1.1 Overview

In this lab, you will create a new project, [HTTPBasicsXX](#), with an orchestration that receives an HTTP request and sends back a hard-coded response.

When the orchestration is complete, you will publish and deploy the project. The orchestration will then be tested using both HTTP GET and POST methods. With the orchestration Logging Level set to “All” you will be able to examine both the headers and body of the HTTP request.

1.2 Concepts

A **Project** is a set of configuration documents that describe the data and components, such as Endpoints and Orchestrations, involved in an integration solution.

Endpoints define the configuration information needed to communicate with external systems or to provide access to the integration appliance’s running orchestrations.

An **Orchestration** is the sequence of processing activities (such as data mapping and control logic activities), endpoints, and data types defined and configured using Studio.

An inbound **HTTP Endpoint** allows the appliance to receive HTTP requests from an external system.

An **HTTP Receive Request** activity may be used as a Starter activity. In actual use cases, this type of starter is independent of polling intervals. As a lightweight, testing tool, it may be used to kick off a process or to perform queries that allow you to test the effect of your orchestrations.

Later, the **HTTP Request-Response model** will be used in conjunction with database operations to 1) return contents of a database table and 2) to insert a new row into the same table.

The **Web Management Console** (WMC) is a browser-accessible tool that allows you to monitor the status of orchestrations, the health of the Appliance and to drill down to detailed information for specific messages, orchestrations or their activities, and any errors that occur.

1.3 Activities Used

- HTTP Receive Request
- HTTP Send Response

1.4 Document Keys

Menu bar: *Edit* → *Preferences*

Studio Artifact: **HTTP Receive Request**

Entered Text: 'Hello Receive Request'

Project Object Name: [SampleProject1](#)

Context Menu Item: [Define Default Value](#)

Activity Checklist Step: [Pick Endpoint](#)

Orchestration Variable: [Account](#)

1.5 Cast Iron Hello

__1. Your student suffix is: _____

__a. During the lab work, project names and artifacts will contain a 'XX'. Replace the 'XX' with your student suffix. **(Please ask your instructor)**

__2. Start the WebSphere Cast Iron Studio. The icon for the WebSphere Cast Iron Studio is located on the desktop.

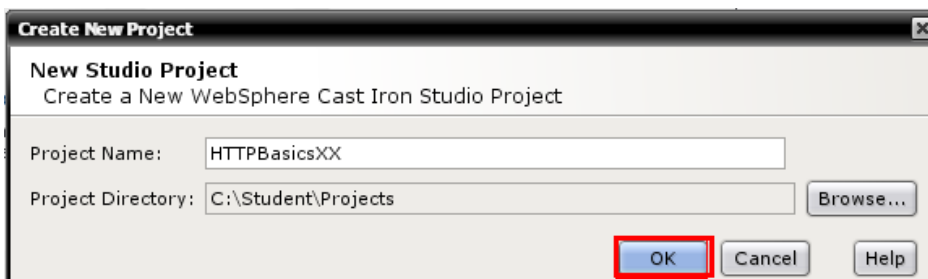


__3. Create a new project

__a. Click *File* → *New Project* or click *Create Project* on the splash screen

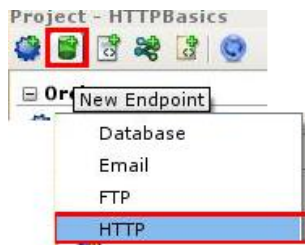
__b. Enter "HTTPBasicsXX", where XX is your student suffix.

__c. Click OK.

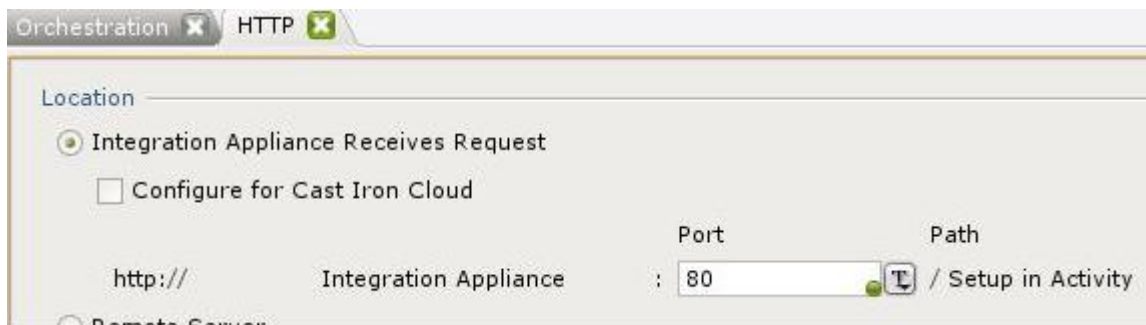


__4. Create a new **HTTP Endpoint**.

__a. Use the New Endpoint button on the Project tab.



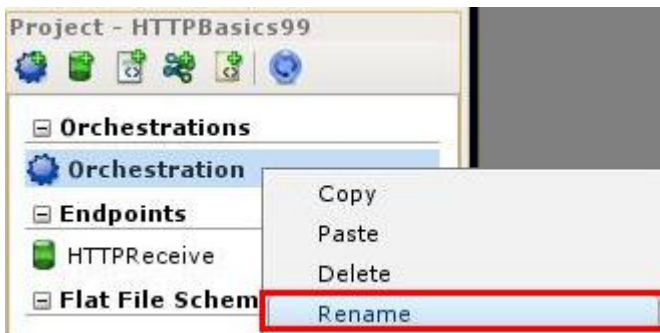
__b. For this **HTTP Receive Request** endpoint the defaults will suffice.



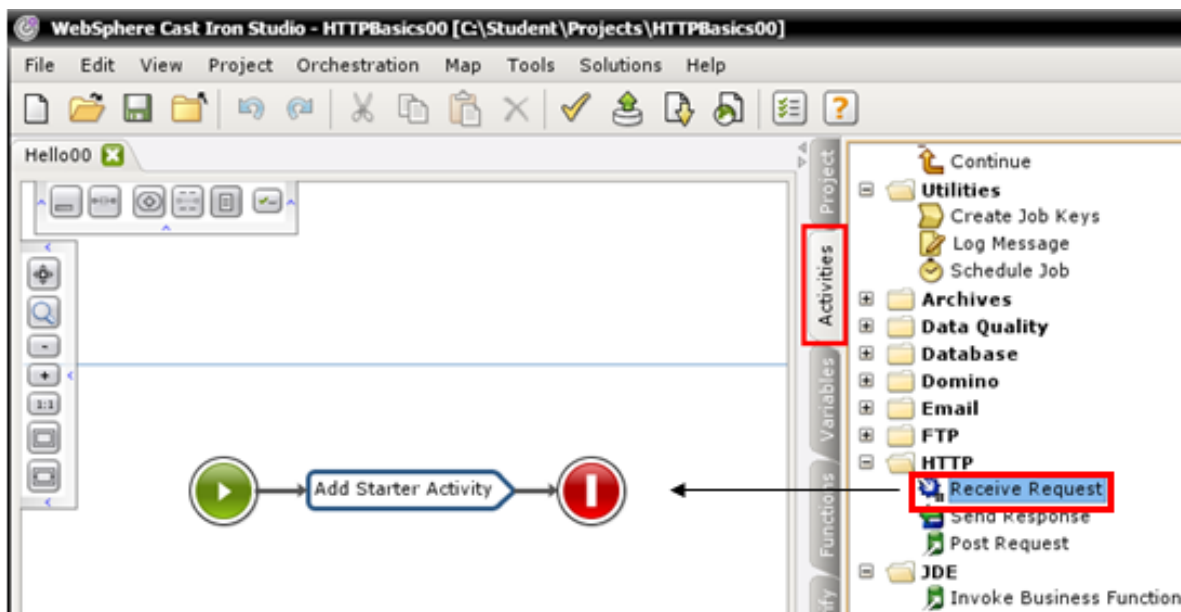
__5. Close the HTTP tab, and then rename the endpoint to **HTTPReceive** by clicking on the name in the Toolbox.



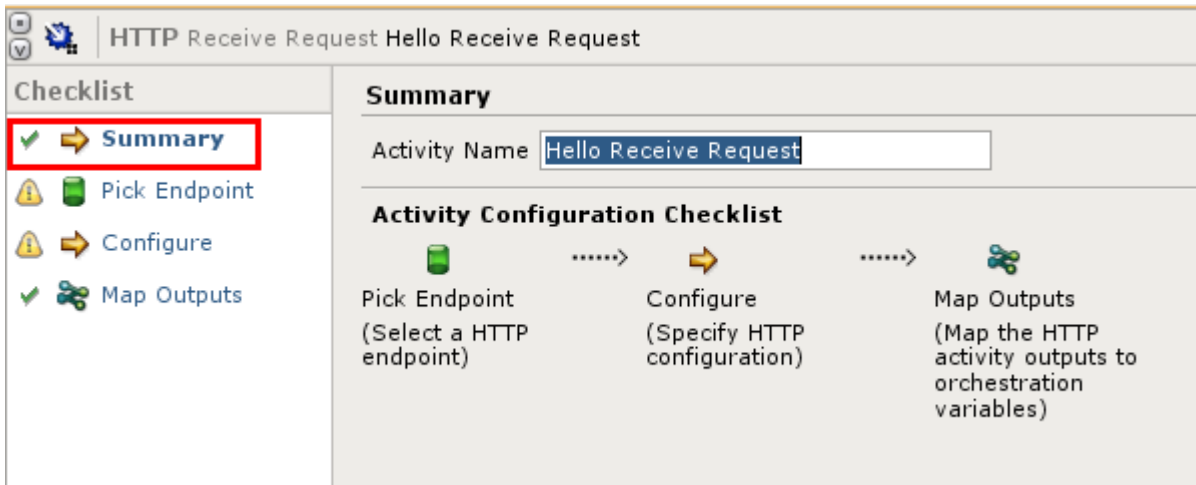
- __6. Rename the default orchestration to **HelloXX**, this time by right-clicking the existing name (as an alternate method).



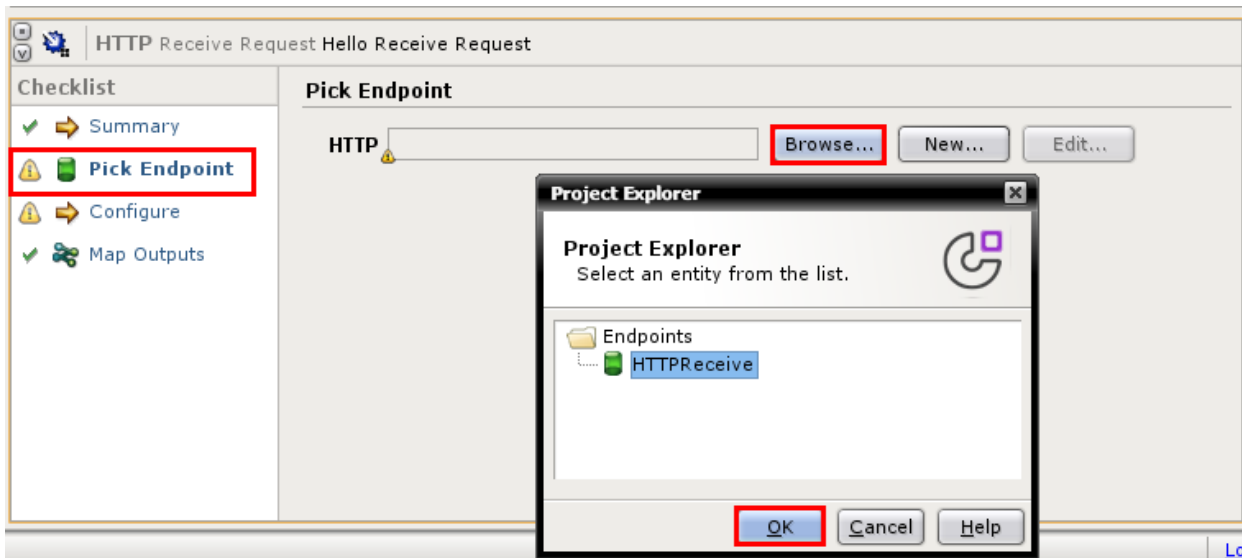
- __7. Start the orchestration by opening the **Activities** tab in the Toolbox and dragging an **HTTP Receive Request** activity on to the workspace. For new activities in sequence you don't have to be precise about where you drop the object. Studio will place it in the orchestration correctly.



- __8. Keep in mind that the first activity in an orchestration must be a starter activity, such as this one. When it's added to the flow, the Properties pane will display the summary information for this activity. Start by renaming this activity to [Hello Receive Request](#).



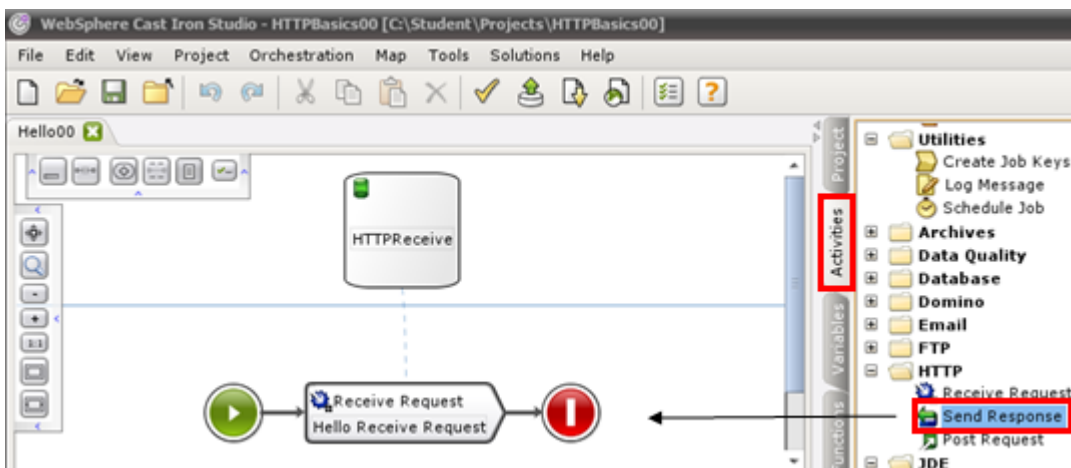
- __a. The checklist graphically shows the configuration steps that need to be completed.
- __b. Select the [Pick Endpoint](#) step and browse for the [HTTPReceive](#) endpoint you defined earlier.



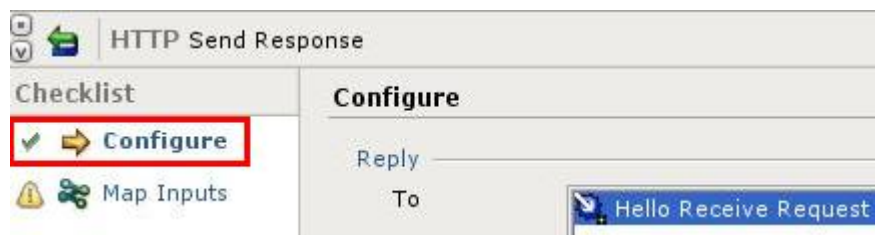
- __c. Select the **Configure** step and enter 'HelloXX' for the listening URL, and also check the option 'Requires a Reply'. For example:



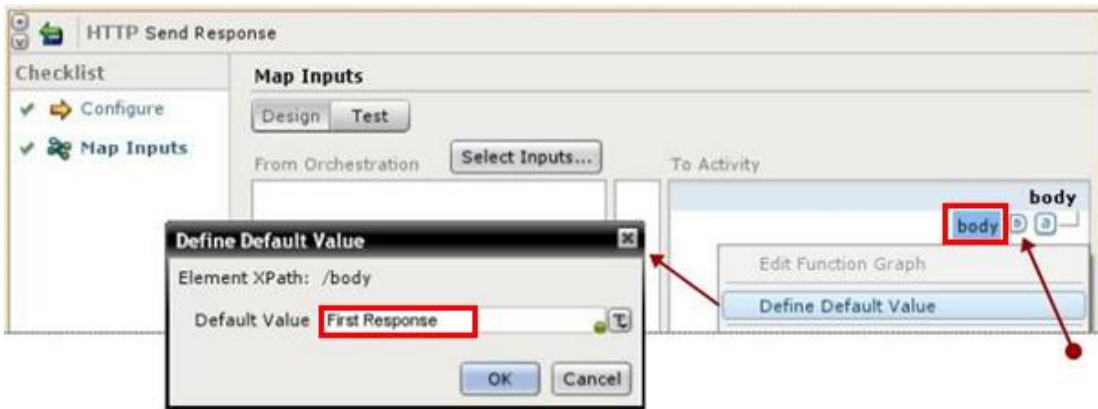
- __9. Add an **HTTP Send Response** activity to the orchestration, by dragging an **HTTP Send Response** activity on to the workspace:



- __a. The **Configure** step should already show that this activity is replying to **Hello Receive Request**.

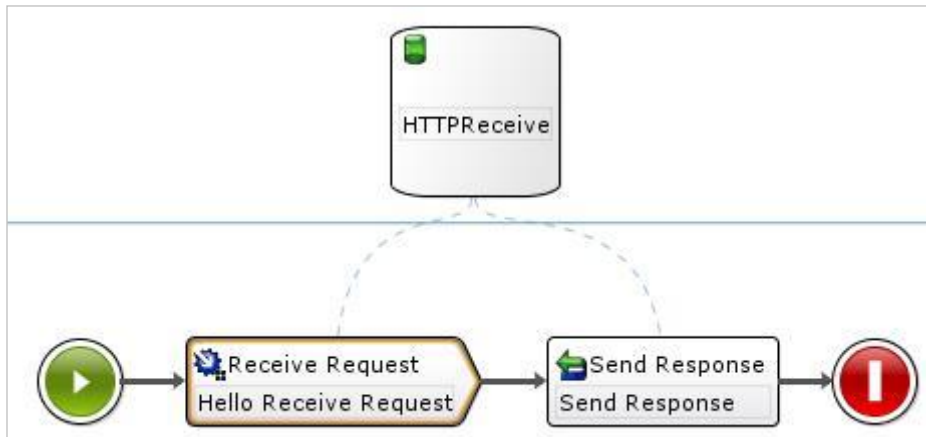


- ___10. Configure the **Map Inputs** step to simply return a fixed value. Right-click on the **body** object to bring up its context menu and select the **Define Default Value** option and type in 'First Response'.

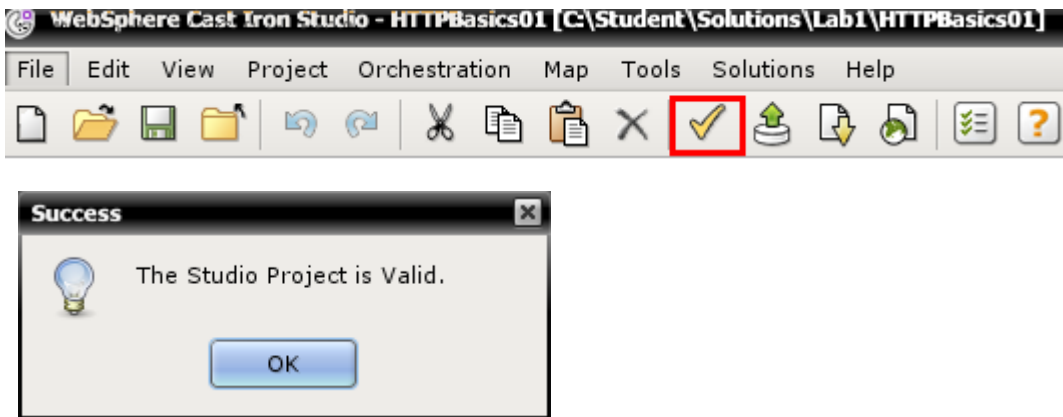


Notice that the body object has a new icon showing it has a Default value

- ___11. The completed orchestration will look like this:



__12. Validate the project via *Orchestration* → *Validate*, or via the Toolbar icon:



__a. Save your project from the Toolbar, using the Toolbar icon:



__13. Publish the project, saving first if prompted, via menu item *File* → *Publish Project*, or via the Toolbar icon:



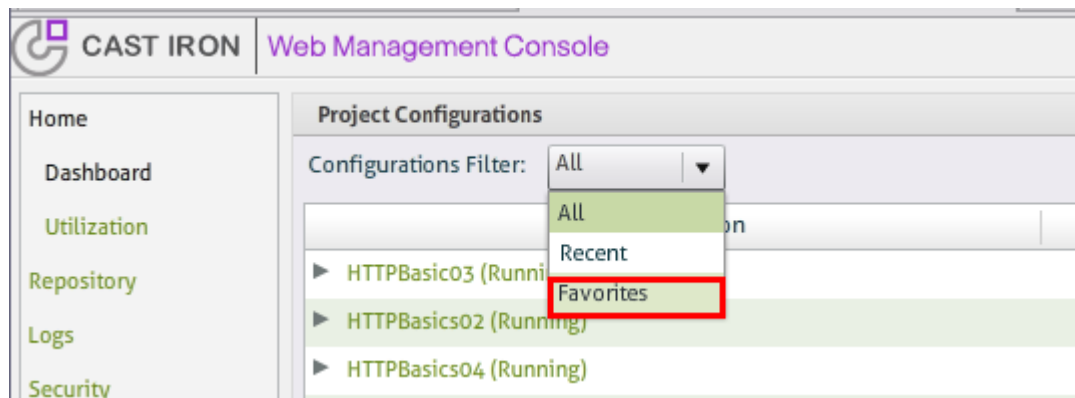
__a. Enter the connection details the first time and click OK. A successful publish will return a confirmation message (Refer to **Connection Parameters Spreadsheet**).

The screenshot shows the "Publish Project" dialog box. The title bar reads "Publish Project". The main text says "Publish Project" and "Publishes a WebSphere Cast Iron Studio Project to an Integration Appliance". There are four input fields: "Host name:" with the placeholder "<Appliance EMgmt Address (WMC)>", "User Name:" with the placeholder "< WMC Id >", and "Password:" with the placeholder "< WMC Password >". Below these fields is a checkbox labeled "Secure" which is checked. At the bottom right, there are "OK" and "Cancel" buttons, with the "OK" button highlighted in red.

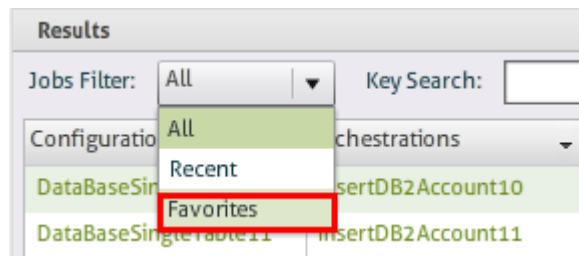
- __14. From the Firefox browser, launch the **Web Management Console** using the URL, **Appliance EMgmt Address (WMC)** provided on the Connection Parameters Spreadsheet.
 - __a. Log in using the, <WMC Id> and <WMC Password> from the Connection Parameters spreadsheet.



- __15. Once in the **Web Management Console**, from the *Home* → *Dashboard*, select the 'ALL' drop-down in Project Configurations and select 'Favorites'. This will limit the view to only your projects.



- __a. Repeat this step in Results, to limit the view to only your projects.



- __16. From the *Home* → *Dashboard* page click the project/configuration name link.

Project Configurations	
Configurations Filter: All	
Configuration	Running
Hello00	0
	0

- __17. Clicking the configuration's link opens the edit page with the list of orchestrations displayed.

- __18. Click the Edit button in the **Orchestrations** section.

Configuration Details

Summary

Configuration: HTTPBasics00 # Orchestrations: 1
 Status: Undeployed # Properties: 0
 Last Published: 05/09/2011 11:31:17 AM # Assets: 0
 # Downtimes: 0

[Download](#)

Orchestrations (2)

Name	Status	Logging Level
Hello00	Enabled	Error Values

[Edit](#)

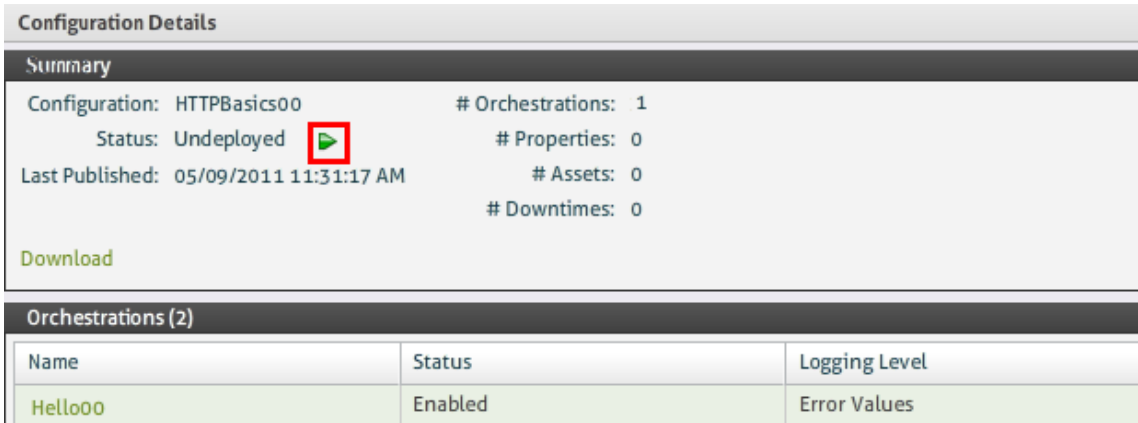
- __19. Apply changes to all the orchestrations in the project with the unnamed top row, or select the orchestrations individually. Change the Logging Level to 'All' and click the Save link.

Edit Orchestration Settings

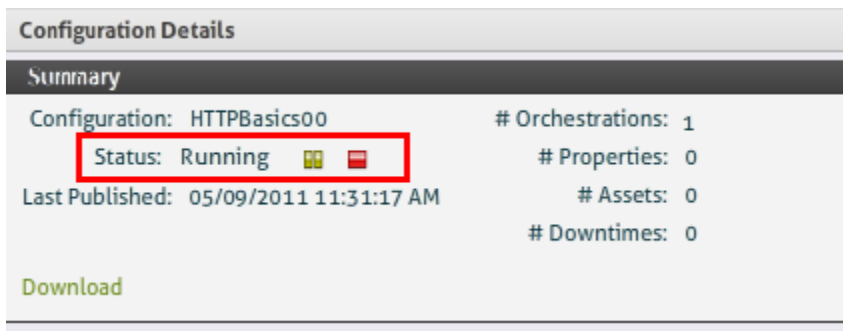
Name	Enabled	Logging Level	Log Synchronously	Max Simultaneous Jobs
	<input type="checkbox"/>	None	<input type="checkbox"/>	<input checked="" type="checkbox"/> Unlimited
Hello00	<input checked="" type="checkbox"/>	All	<input type="checkbox"/>	<input type="checkbox"/> Unlimited 10

[Save](#) [Cancel](#) [Help](#)

__20. Back on the Configurations page, deploy and run the project using the Run Configuration icon:



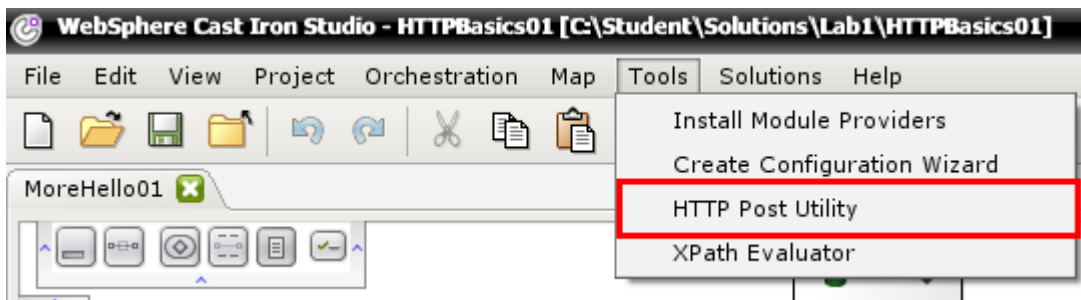
__21. The icons change to show the status of the configuration.

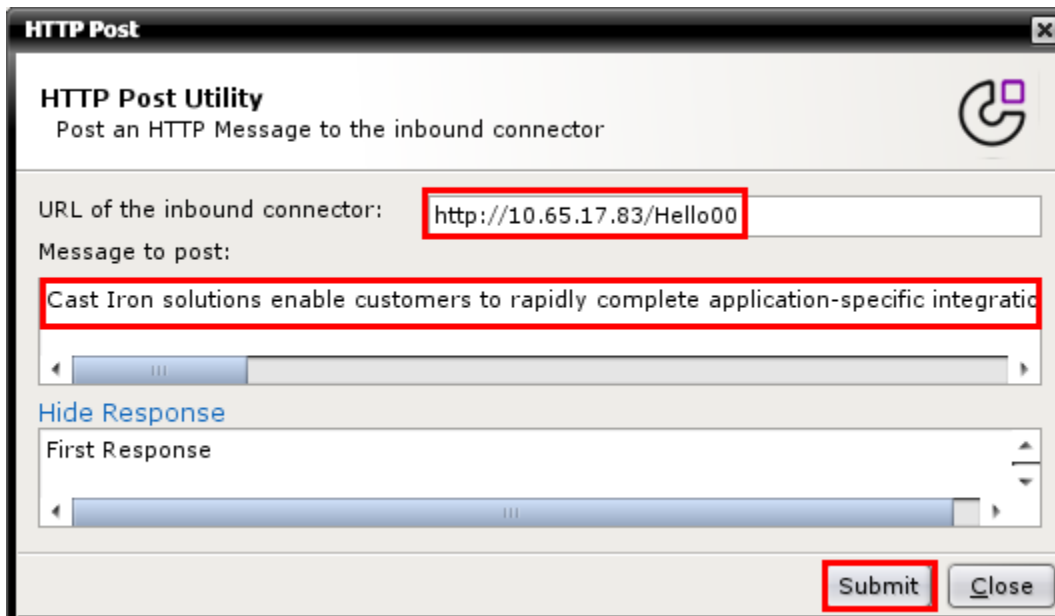


__22. Trigger the orchestration twice, once for POST and once for GET. To issue a POST, use the HTTP Post Utility that is included with the Studio installation. To issue a GET, simply enter the URL into your browser. In either case, use the IP address of the appliance's **Edata** interface. See the Connectivity Parameter spreadsheet.

__a. POST: Launch the HTTP Post Utility with the appropriate URL. Then browse for the input file in the **C:\Student\Lab Resources\Lab 1** folder. After submission you should see the reply text that was entered into the **body** object of the **HTTP Send Response** activity.

HTTP Post Utility from the Toolbar



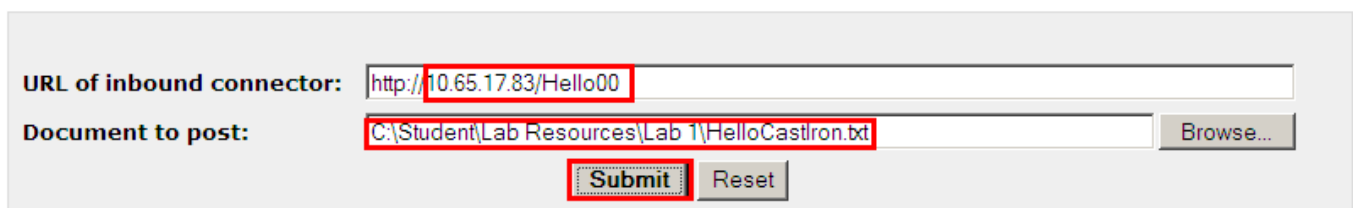


Or

HTTP Post Utility from Start\All Programs\IBM\WebSphere Cast Iron Studio







Post a document to the WebSphere Cast Iron Integration Orchestration



First Response

- ___23. The WMC will show the results of each run. From the Dashboard, expand the configuration and click on the orchestration being tested. This will display the Orchestration Details page that shows the results of each run of the orchestration.

Configuration	Running	Completed	Errored	Total	Actions
▼ Lab1_HTTPBasics99 (Running)	0	1	0	1	   
Hello00 (Running)	0	1	0	1	
InputHello00 (Running)	0	0	0	0	

Orchestration Details			
Configuration: HTTPBasics00			
Orchestration: Hello00			
Key/Job ID	Status	Start Time	End Time
94D2DFA4B9F52B7616FD71354F1F4041	Completed in 0.01s	05/09/2011 01:25:26 PM	05/09/2011 01:25:26 PM

- ___a. Select the Job ID link for the POST run in the top pane. Click the 'Details' link in the bottom pane and then the **HTTP Send Response** section. Click on the 'body (input)' link to dynamically generate a file with the variable's contents
- ___b. This will open a file containing the response that was displayed in the browser
- ___c. Subsequent labs will present better opportunities for viewing input and output values

Job Details - 96923CECE8BA01B51C7D6FCBE166460B

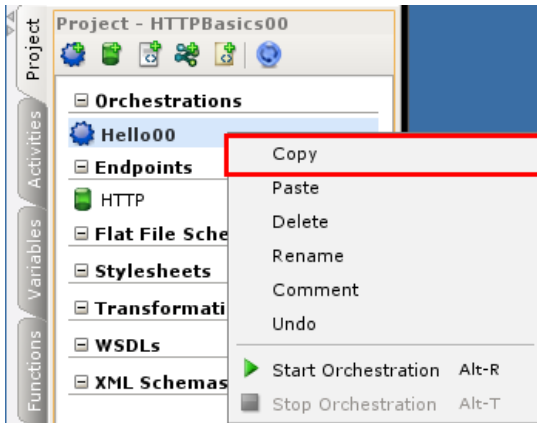
Summary Job Keys **Details**

Displaying variables and events for activity: HTTP - Send Response

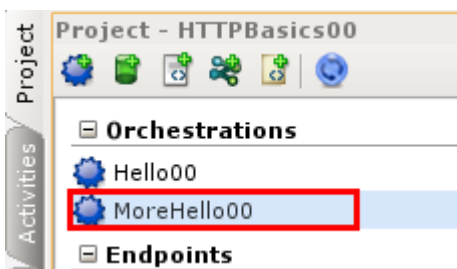
Elapsed	Activity	Variable
0.043s	 HTTP Hello Receive Request	Your content is ready. Download Now 
0.017s	 HTTP Send Response	body (input)

1.6 Enhanced Cast Iron Hello

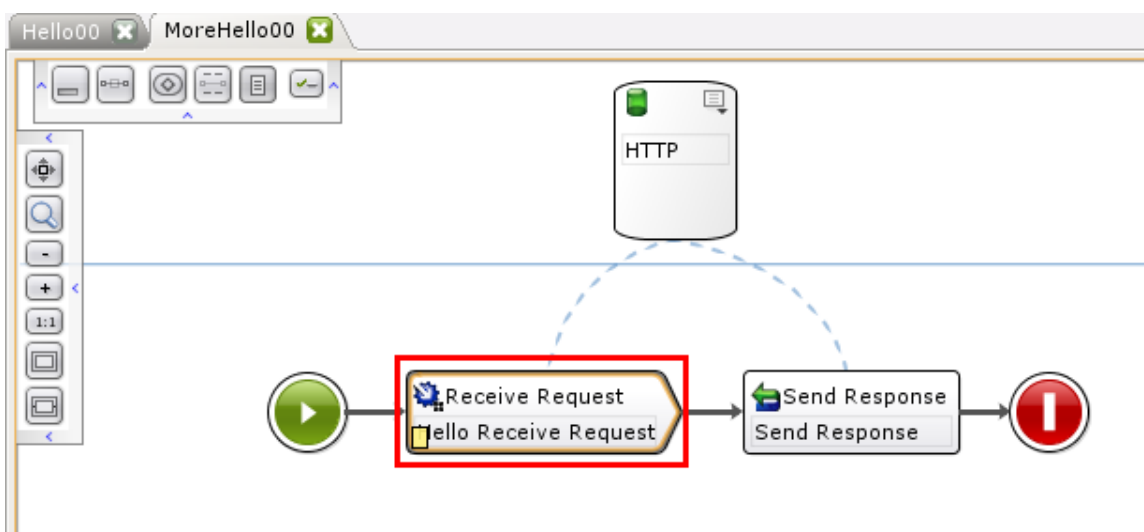
1. Right-click on the orchestration to Copy and Paste a new orchestration from your existing orchestration.



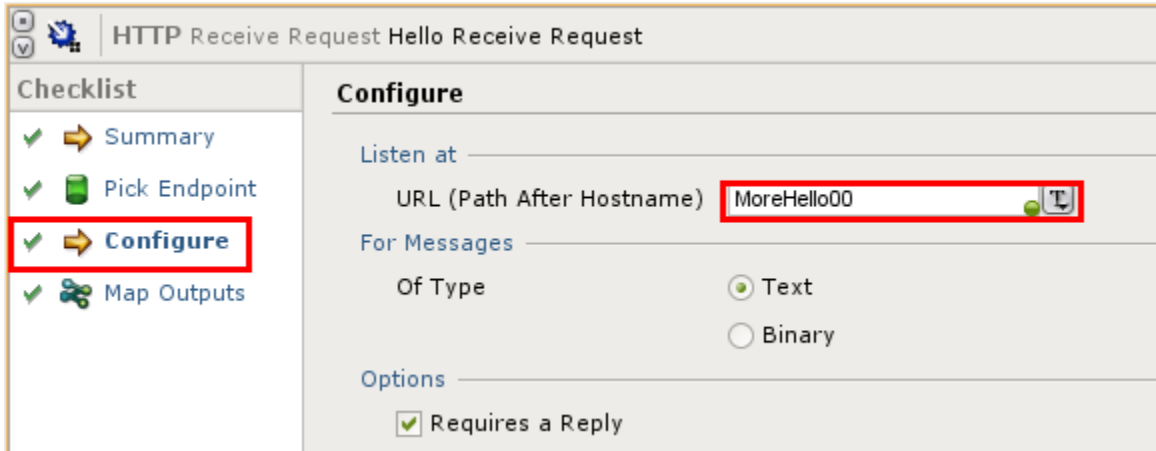
2. Rename the copied orchestration to MoreHelloXX.



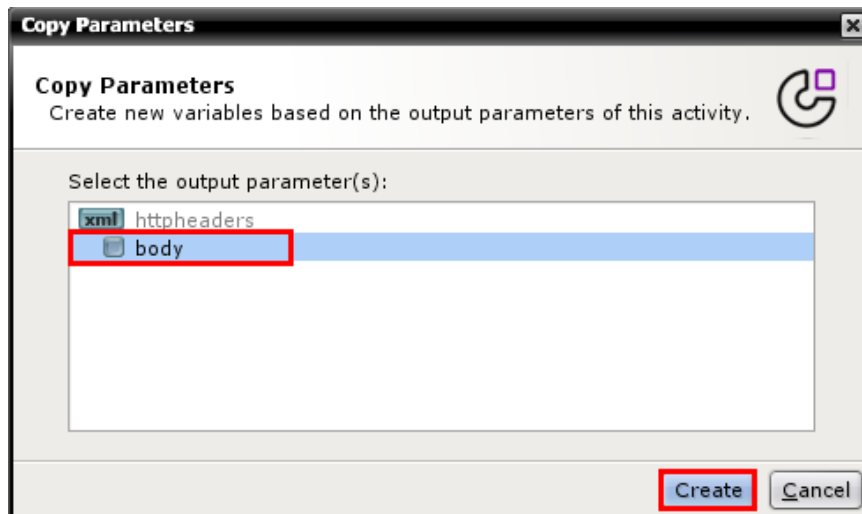
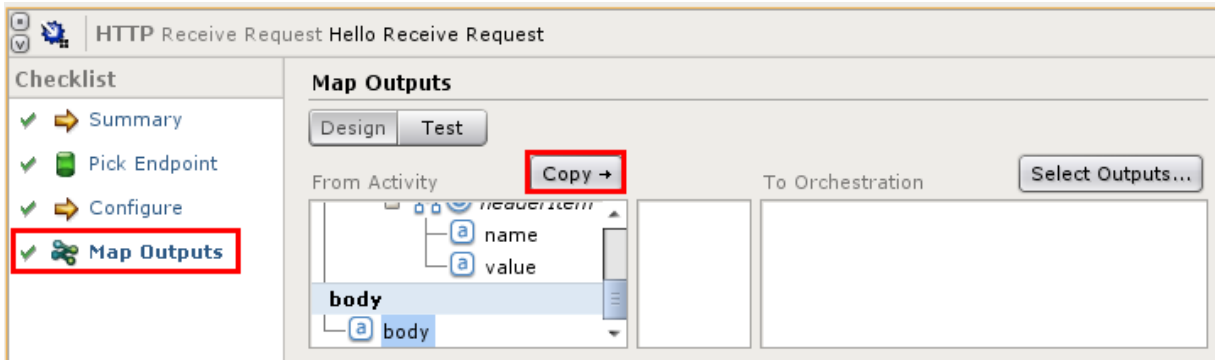
3. Highlight the **Receive Request** activity on the workspace.



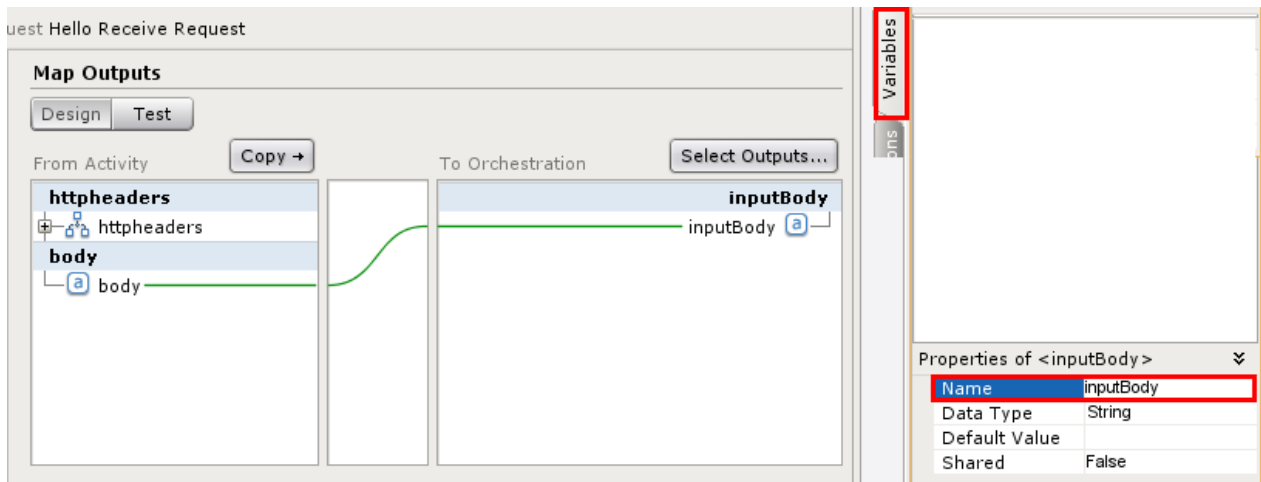
__a. Select [Configure](#), from the Checklist and change the URL to MoreHelloXX.



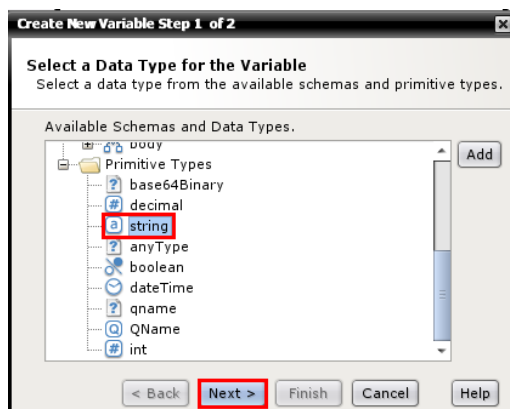
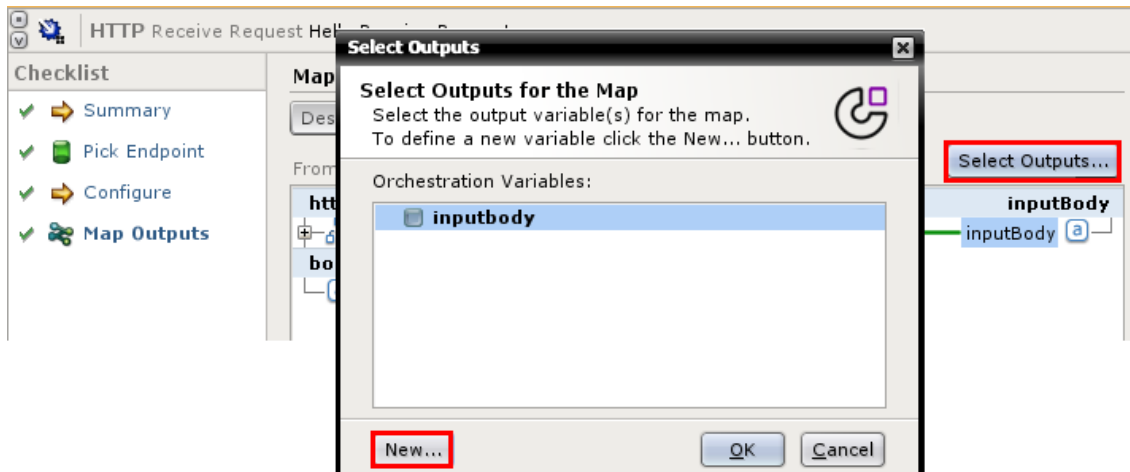
__b. [Map Outputs](#): Copy input **body** to orchestration, click on the **body** and select Create.



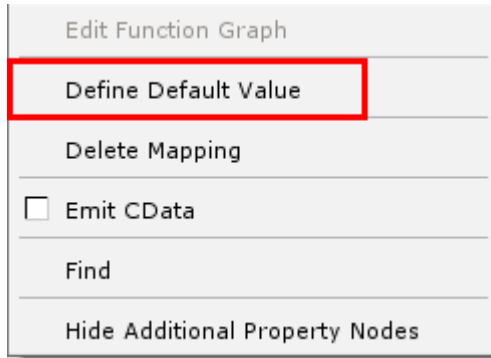
- __4. Select the 'Variables' Tab and rename the orchestration variable to **inputBody**



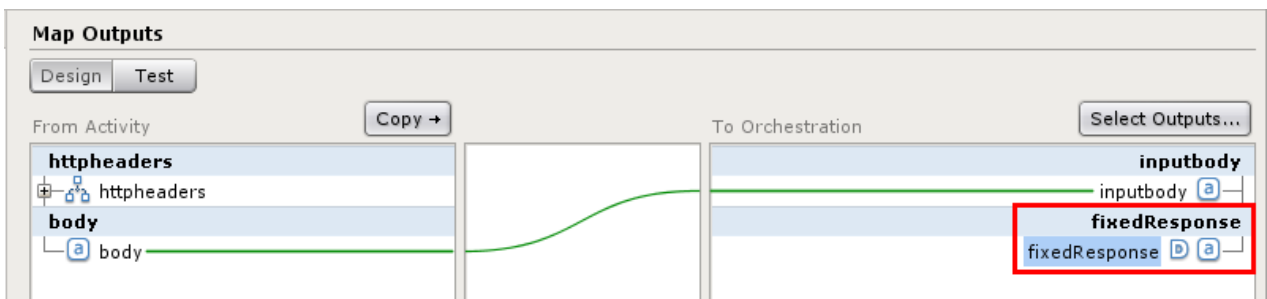
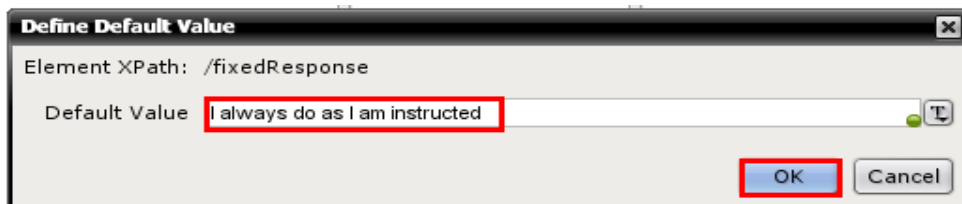
- __5. Manually create a new string variable with a default value of your choosing using the **Select Outputs** and **New** buttons.



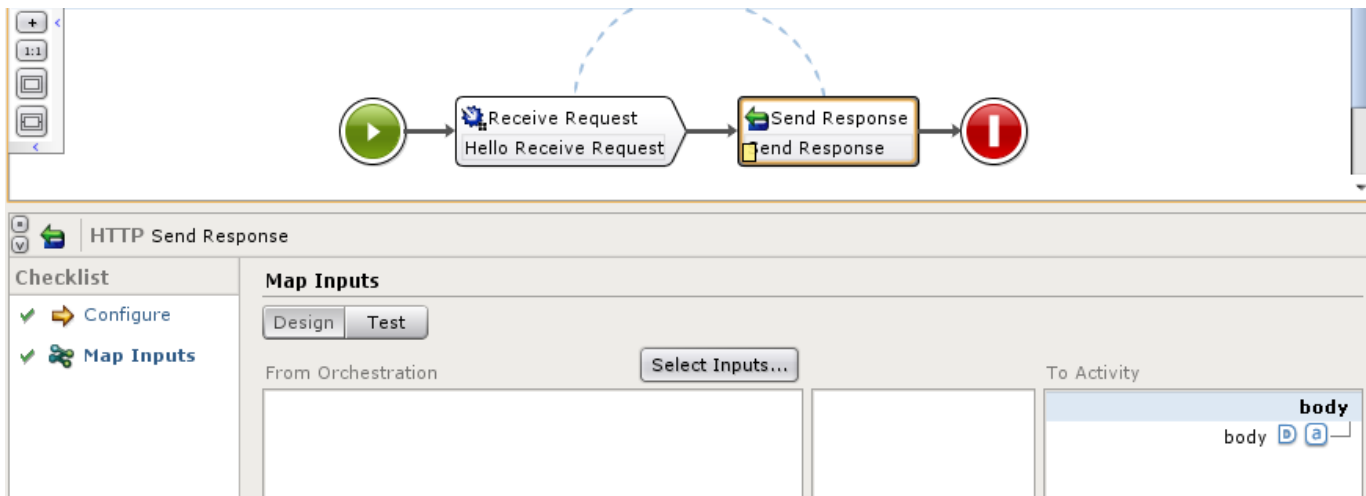
__a. Right Click, on 'fixedResponse' and define a default value.



__b. Add the default text: ex: 'fixedResponse' = I always do as I am instructed



__6. Highlight the **Send Response** activity in the workspace.

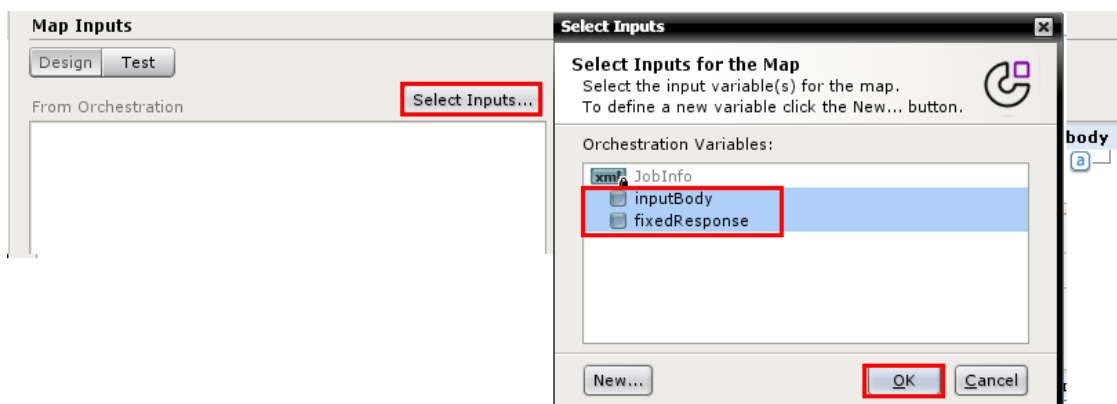


__a. Select the Map Inputs and delete default mapping for **body** (context menu of object).

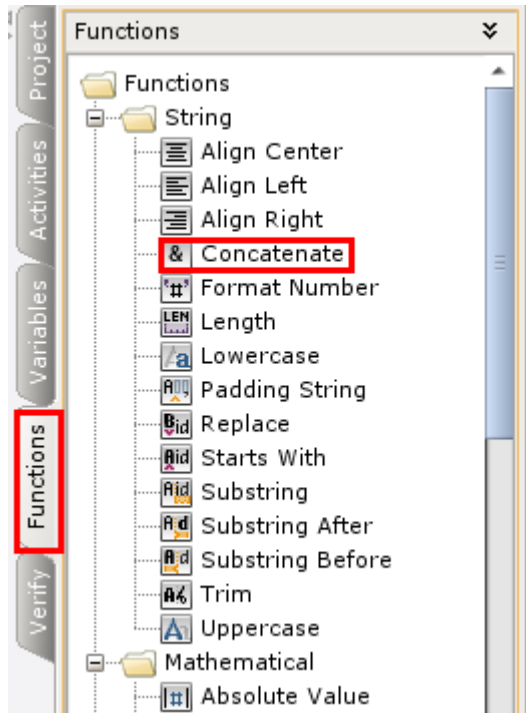


__b. Click '**Select Inputs**' button.

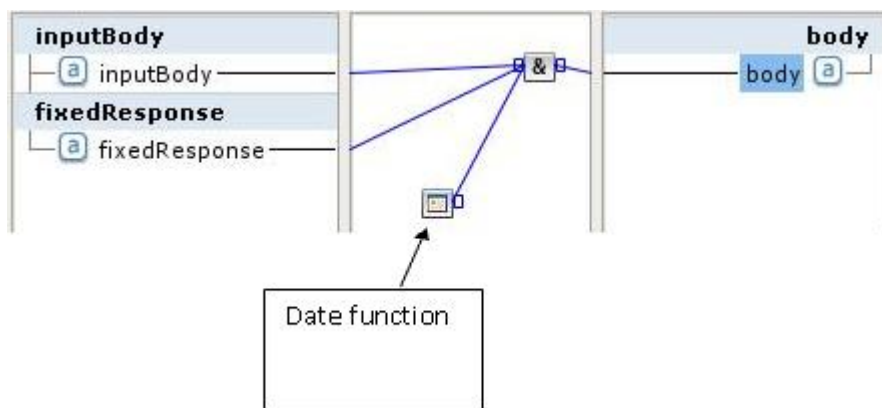
__i. Select both variables from steps 4 and 5.



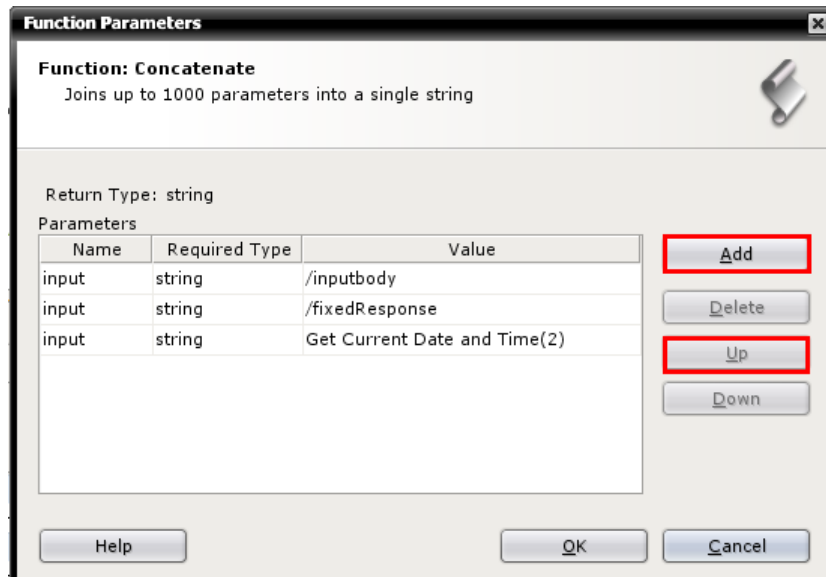
- __c. Select the Functions Tab and drag the '**Concatenate**' function to the function graph (center pane).



- __d. Drag **Get Current Date and Time** function to the function graph pane.
- __e. Connect the input variables and the **Date** function to the **Concatenate** function.

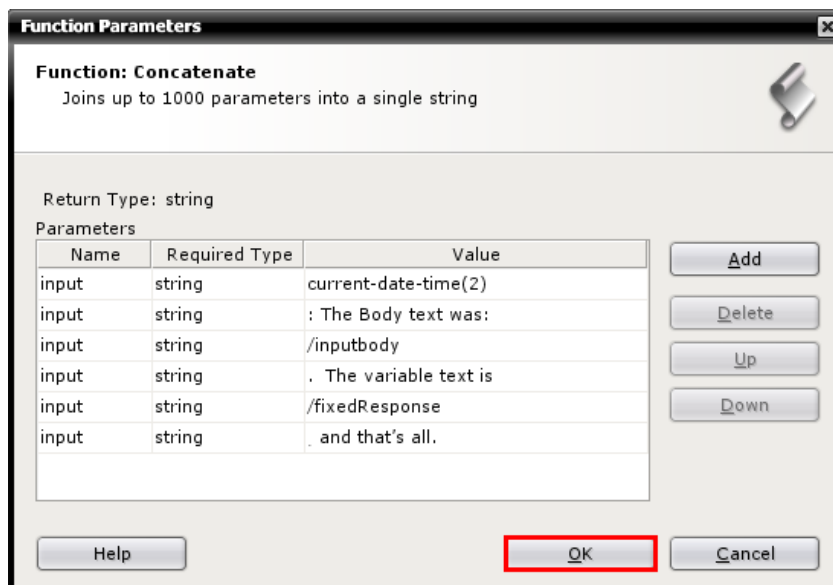


- __i. Double-click **Concatenate** function to open its parameters, then add strings with the **Add** button.

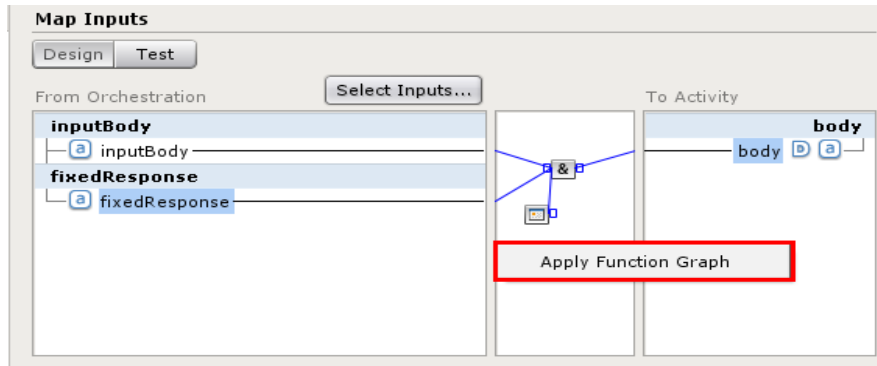


- __ii. Add the following text strings and reorder the parameters to form a coherent output string:

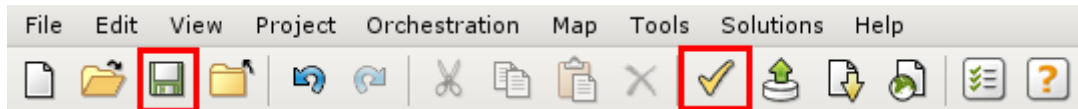
: The Body text was:
. The variable text is
and that is all.



- __f. Map **Concatenate** function to output **body**.
- __g. Right-click in function graph pane to **Apply Function Graph**.






- __h. Save and Verify the project.






- __i. Before you can deploy and test the updated project, you must go out to the WMC, Stop and Undeploy the current project.

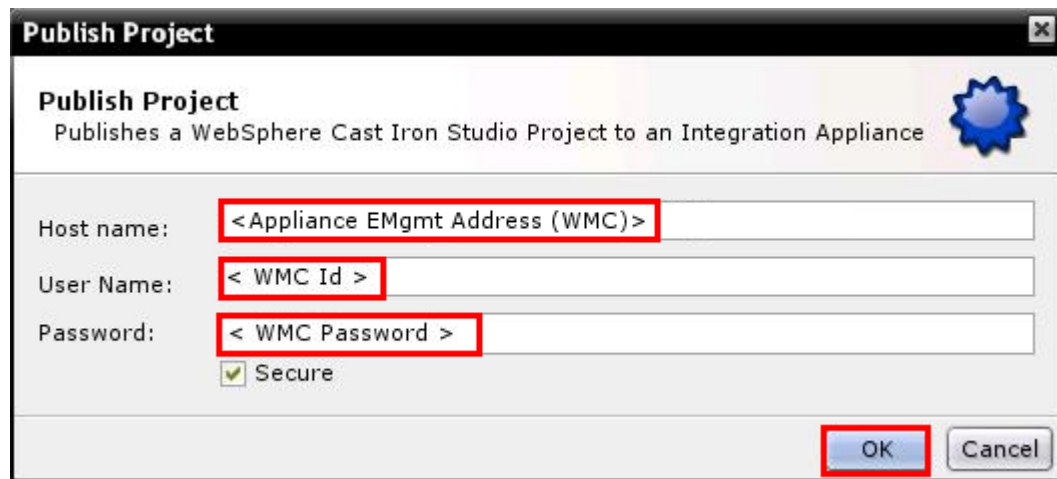
Stop Configuration

Project Configurations							
Configurations Filter: Favorites							
Configuration	Last Published	Running	Comple...	Errored	Total	Actions	
▶ HTTPBasics00 (Running)	05/09/2011 01:11:33 PI	0	0	0	0	  	

Undeploy Configuration

Project Configurations							
Configurations Filter: Favorites							
Configuration	Last Published	Running	Comple...	Errored	Total	Actions	
▼ HTTPBasics00 (Stopped)	05/09/2011 01:11:33 PI	0	2	0	2	  	
Hello00 (Stopped)		0	2	0	2		

__c. Deploy and run the updated project using the Run Configuration icon.



▼ HTTPBasics00 (Undeployed)	0	0	0	0	0				
Hello00 (Undeployed)	0	0	0	0	0				
MoreHello00 (Undeployed)	0	0	0	0	0				
▼ HTTPBasics00 (Running)	0	6	0	6	6				
Hello00 (Running)	0	0	0	0	0				
MoreHello00 (Running)	0	6	0	6	6				

__d. Test the project. Verify and launch the HTTP Post Utility with the appropriate URL using input file in the **C:\Student\Lab Resources\Lab 1** folder just like you did in Part 1 of this lab but just substituting the HTTP endpoint.



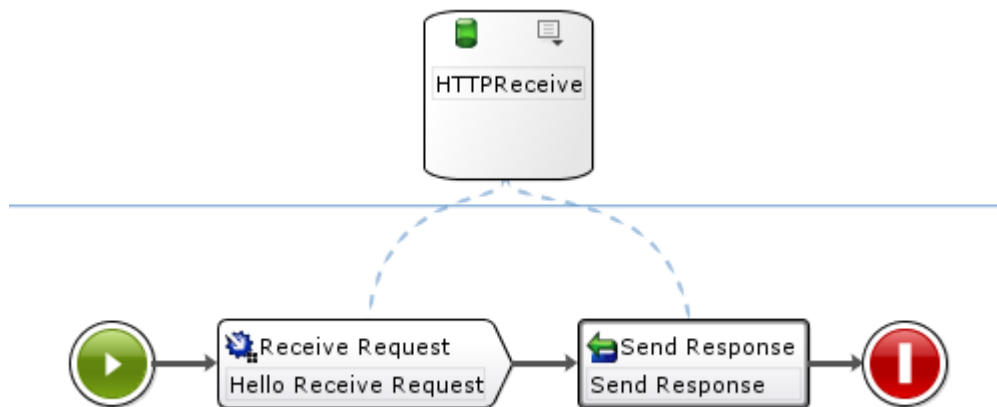


Post a document to the WebSphere Cast Iron Integration Orchestration

URL of inbound connector:

Document to post:

2011-05-25T14:14:14+00:00: The Body text was: Cast Iron solutions enable customers to rapidly complete application-specific integrations using a "configuration, not coding" approach. By using a pre-configured template, rather than starting from scratch with complex software tools and writing lots of code, enterprises complete business-critical projects in days rather than months. Large and midsize companies in a variety of industries use Cast Iron solutions to solve their most common integration needs. The variable text is: I always do as I am instructed. and that is all.



Lab 2 Single Table Database

2.1 Overview

In this lab, you will create a new project, [DatabaseSingleTable](#), with three orchestrations. One uses the functions Substring Before and Substring After in the mapping function to parse an HTTP request string in order to insert that data into a database table. The second orchestration queries the database to return all of the current rows in the table.

2.2 Concepts

In order to access a database, you must create a **Database Endpoint** to provide database connection parameters. These endpoint definitions will be used by Database Activities that fetch data from the database or write data into the database.

Studio provides the ability to create **Configuration Properties** or name/value pairs in a project. The configuration properties parameterize information, such as the connection information that you specify for Endpoints.

The functions can be used to provide capabilities natively available.

2.3 Activities Used

16. HTTP Receive Request
17. HTTP Send Response
18. Database Insert Rows
19. Database Execute Query
20. Transform Write XML

2.4 Document Keys

Menu bar: *Edit → Preferences*

Studio Artifact: **HTTP Receive Request**

Entered Text: 'Hello Receive Request'

Project Object Name: [SampleProject1](#)

Context Menu Item: [Define Default Value](#)

Activity Checklist Step: [Pick Endpoint](#)

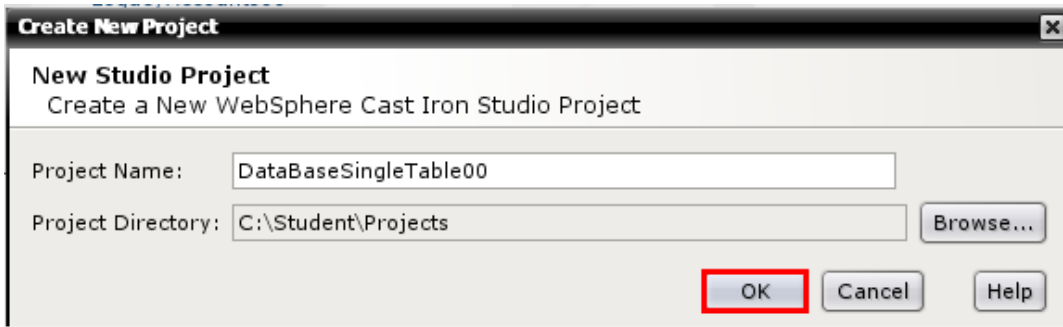
Orchestration Variable: [Account](#)

Function Name: [Substring Before](#)

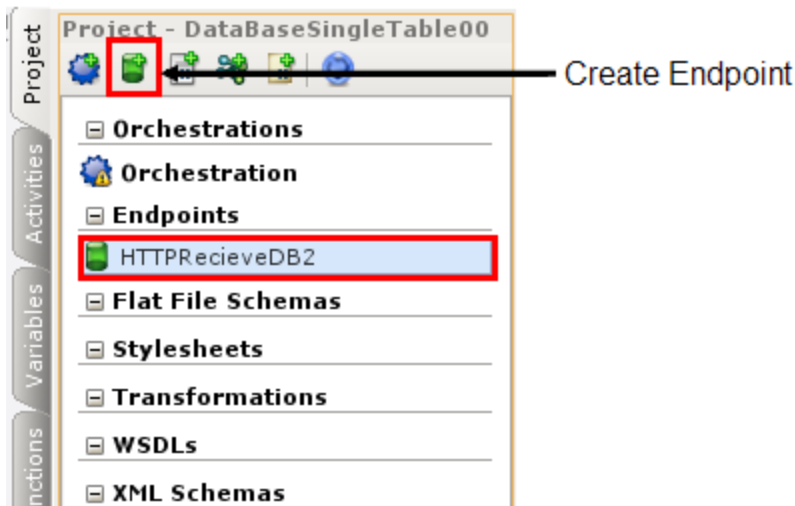
Function Name: [Substring After](#)

2.5 Insert Single Account into Database

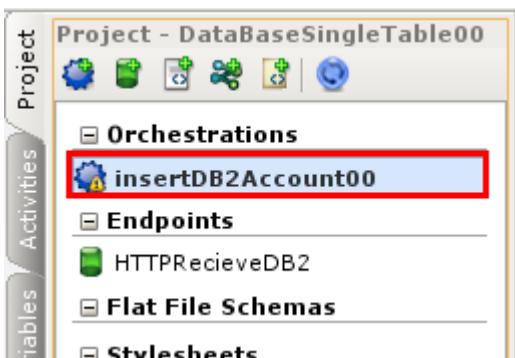
1. Create a new project [DataBaseSingleTableXX](#).



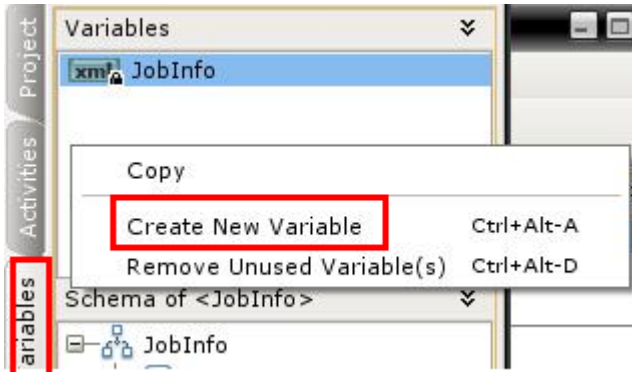
2. Create an inbound HTTP endpoint and rename it to: [HTTPReceiveDB2](#).



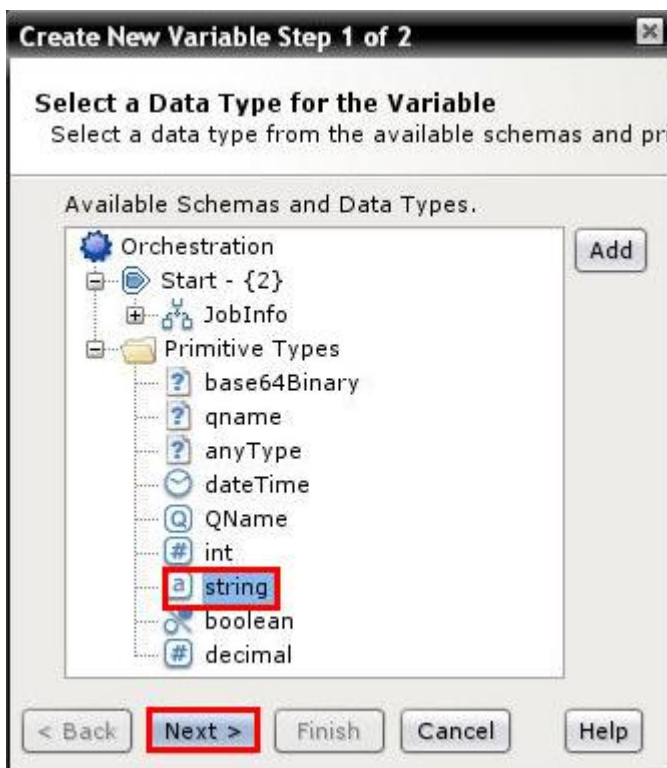
3. Rename the default orchestration to [insertDB2AccountXX](#).



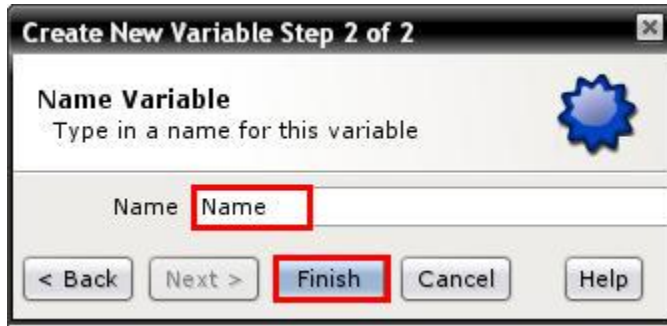
- __4. Create 3 string variables: **Name, Description, Phone**
- __a. Click the Variables tab where you will see the default **JobInfo** variable
- __b. Right-click in the tab's workspace to open the create window (or Ctrl+Alt-A)



- __c. Select the 'string' option and click 'Next'.

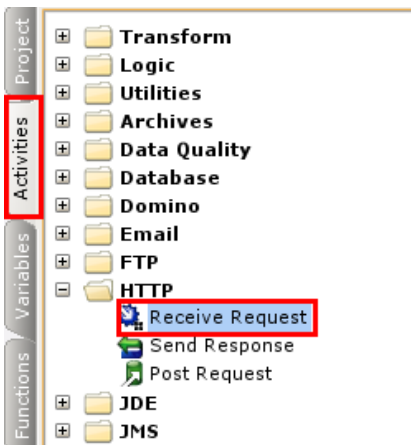


__d. Provide the new variable **Name** and click Finish.

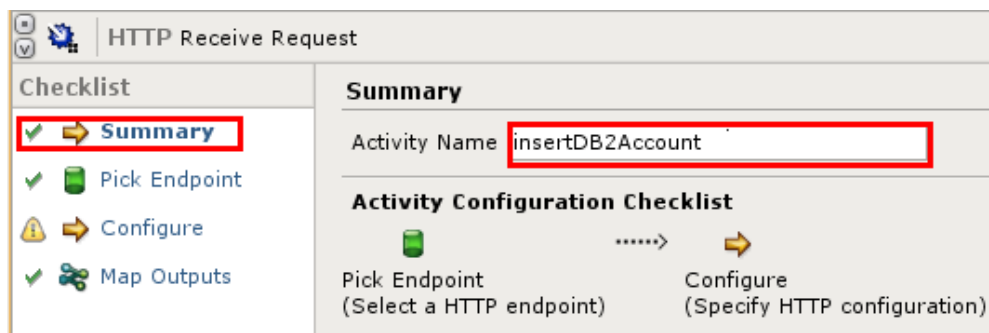


__e. Repeat the previous bullet items to create variables **Description** and **Phone**.

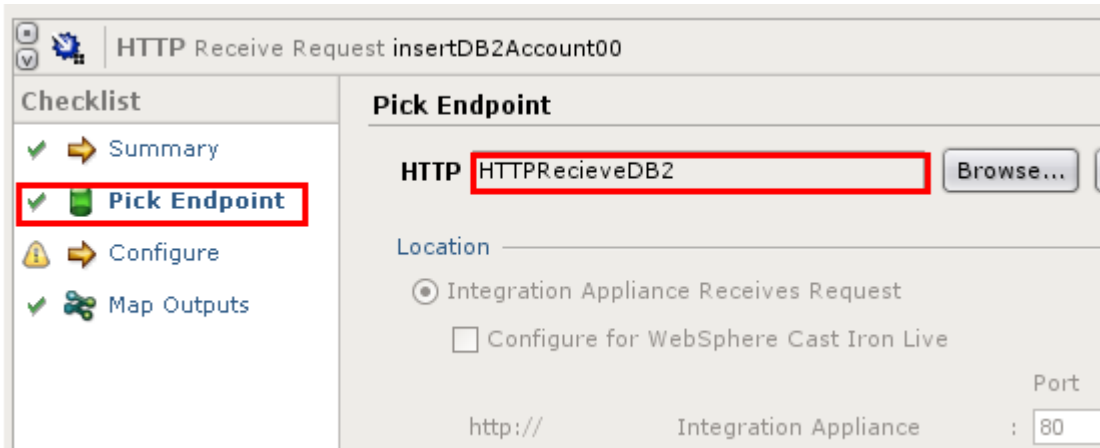
__5. Add a new **HTTP Receive Request** from the Activity Tab to the orchestration.



__6. Renaming it to **insertDB2Account**.

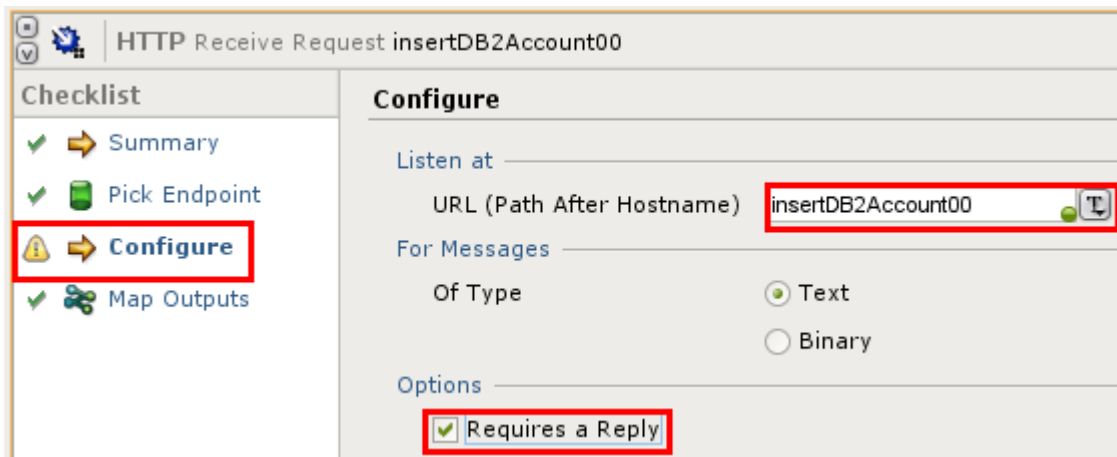


- __a. **Pick Endpoint:** Browse and select the **HTTPReceiveDB2** endpoint.



- __b. **Configure:**

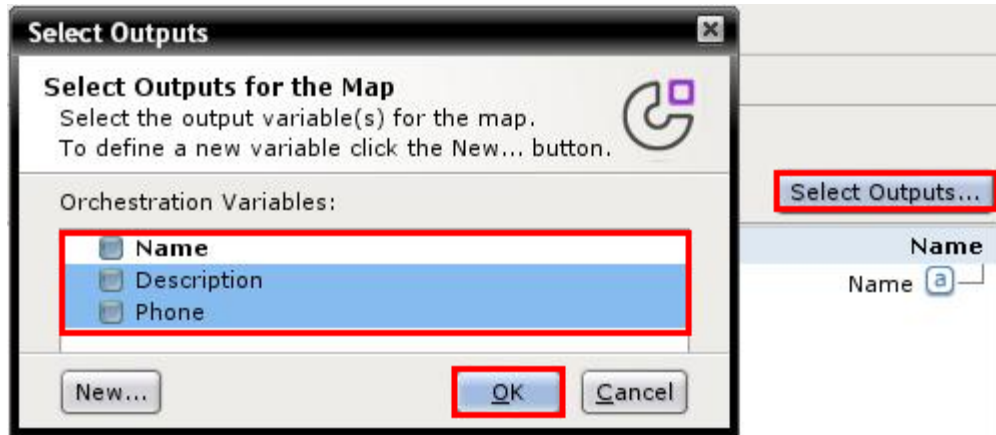
- __i. URL = 'insertDB2AccountXX'
 __ii. Select option 'Requires a Reply'.



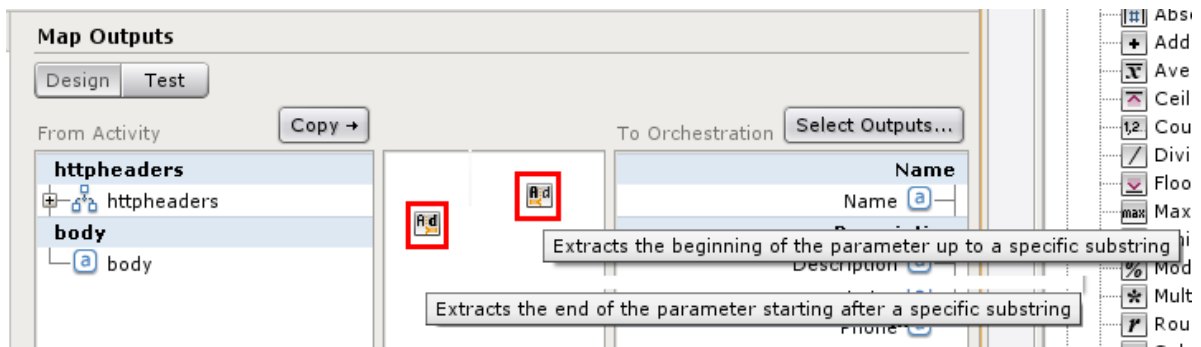
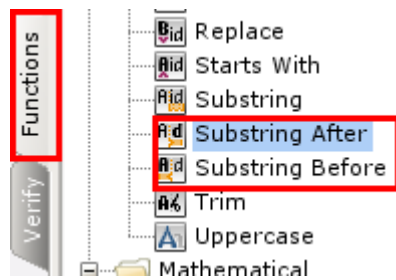
- __7. The input string will look like this, the Substring Afer and Substring Before functions will extract the parameters and values:

Name=CastIron&Description=sometext&Phone=770+555+1234&

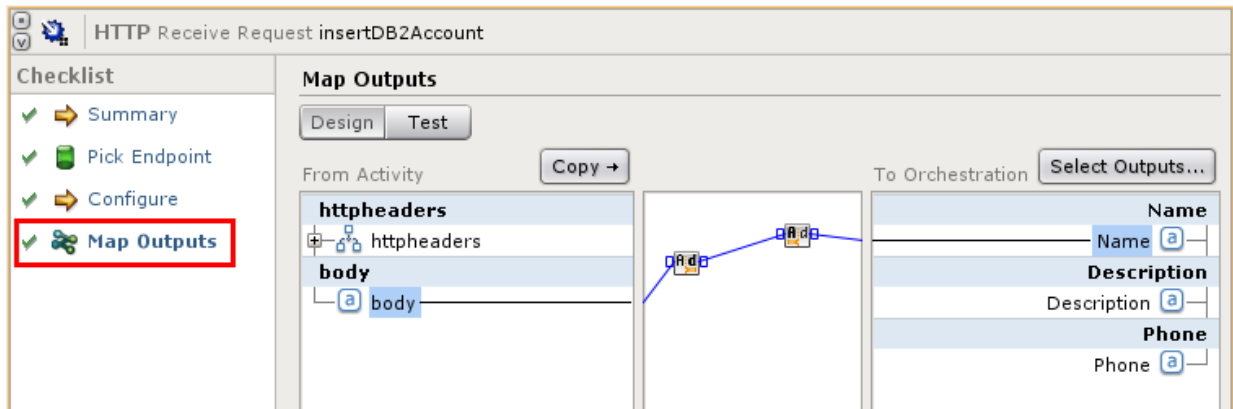
- __8. Set up variables to store the parts of the input string
 - __a. In the [Map Outputs](#) step of the **HTTP Receive Request**, use the Select Outputs button to add the three new variables to the output pane (you can select multiple variables at a time).



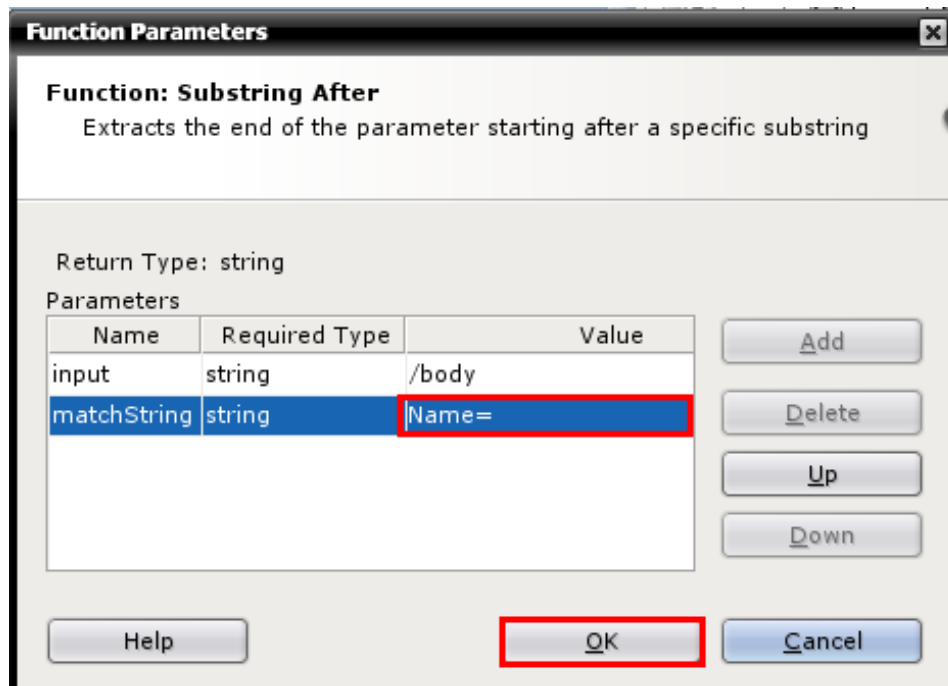
- __9. Select the function Substring after and drag it on to the graph pane. Next select the function Substring Before and drag it on to the graph pane.



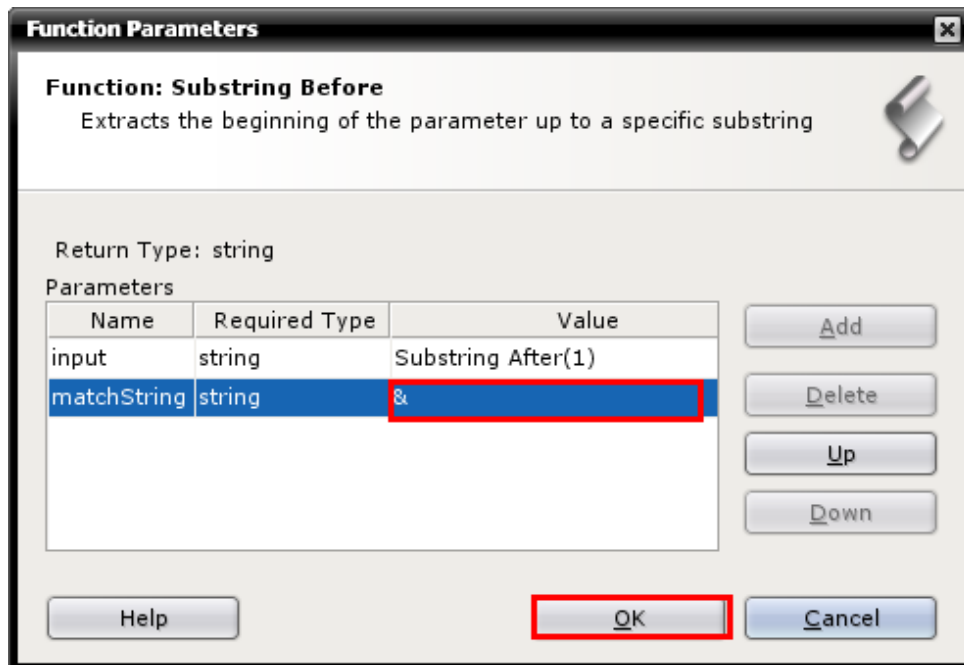
- __a. Drag the input **body** to the function Substring After. Drag the function Substring After to the function Substring After. Drag the function Substring Before to the output orchestration variable **Name**.



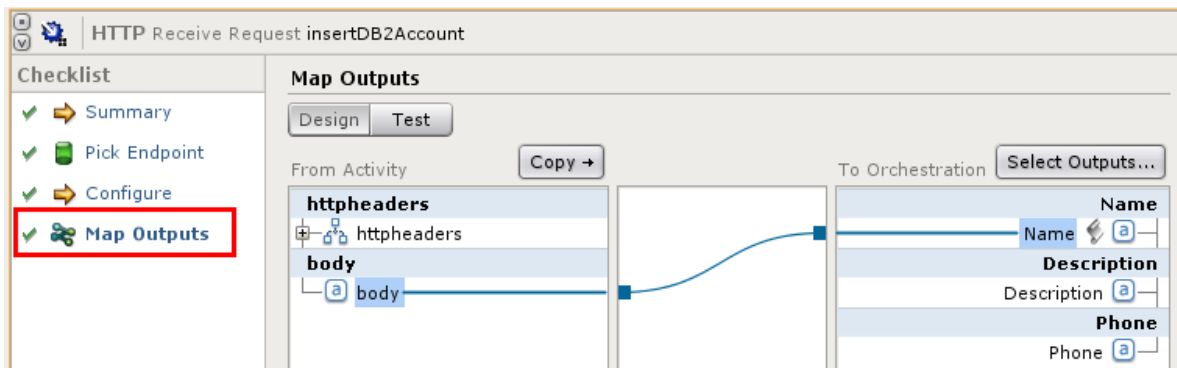
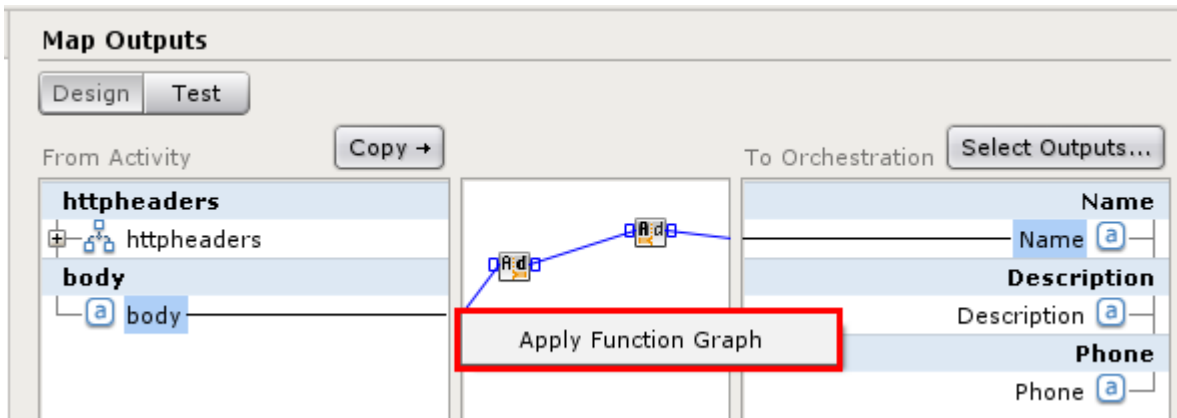
- __b. Double-click the function Substring After and add the 'matchString' value 'Name='.



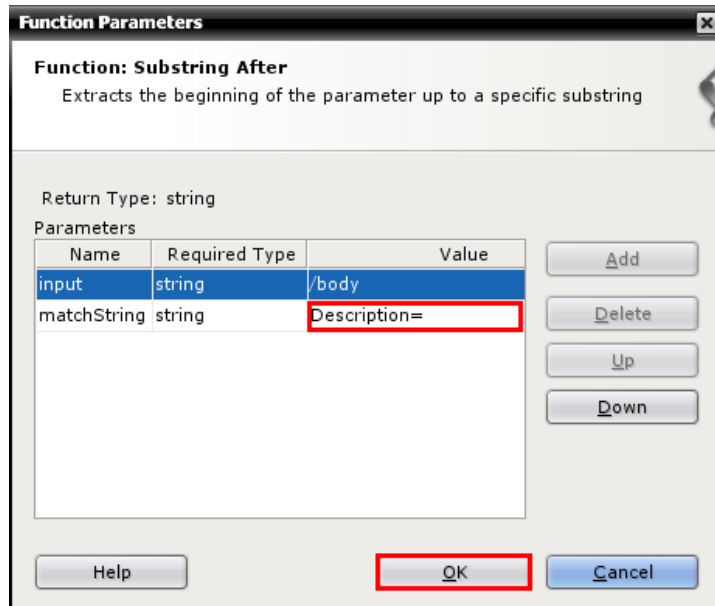
- c. Double-click the function Substring Before and add the 'matchString' value '&'.



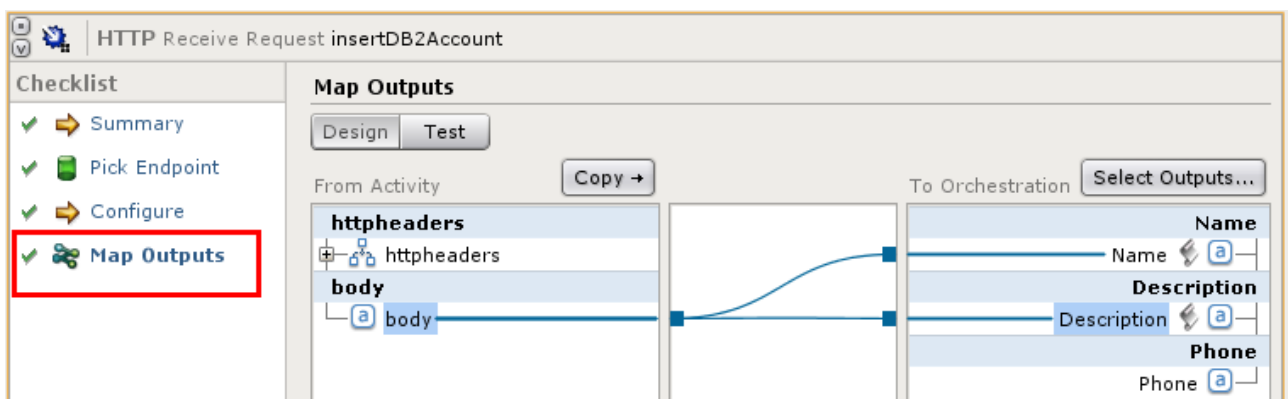
- d. Right-click the graph pane and Apply Function Graph.



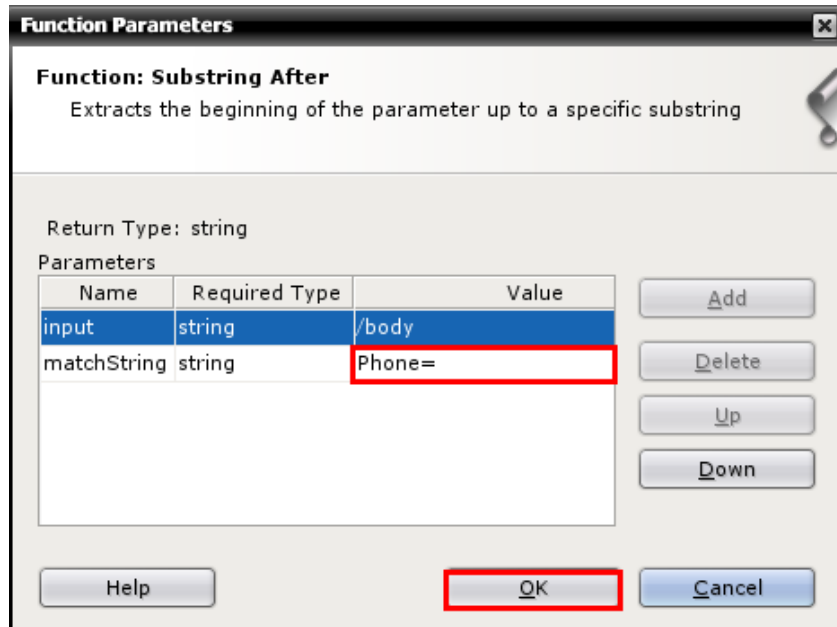
- ___10. Repeat Step 9 two more times, to build the input values for the 'Description' and 'Phone' orchestration variables.
- ___a. Drag the function Substring After and the function Substring Before on to the graph pane.
 - ___b. Drag the input **body** to the function Substring After. Drag the function Substring After to the function Substring Before. Drag the function Substring Before to the output orchestration variable **Description**.
 - ___c. Double-click the function Substring After and add the 'matchString' value 'Description='.



- ___d. Double-click the function Substring Before and add the 'matchString' value '&'.
- ___e. Right-click the graph pane and Apply Function Graph.

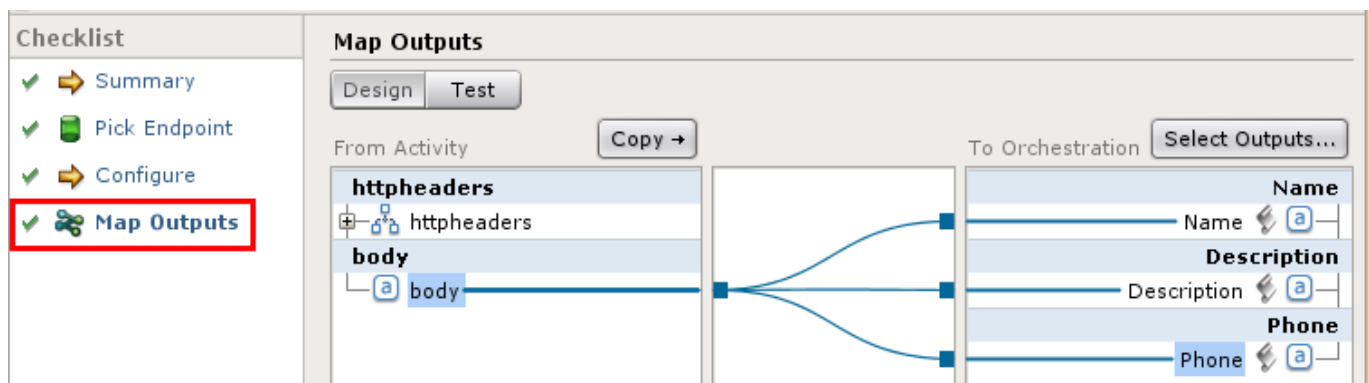


- __11. Repeat these steps one more time for the Phone orchestration variable.
 - __a. Drag the function Substring After and the function Substring Before on to the graph pane.
 - __b. Drag the input **body** to the function Substring After. Drag the function Substring After to the function Substring Before. Drag the function Substring Before to the output orchestration variable Phone.
 - __c. Double-click the function Substring Before and add the 'matchString' value 'Phone='.



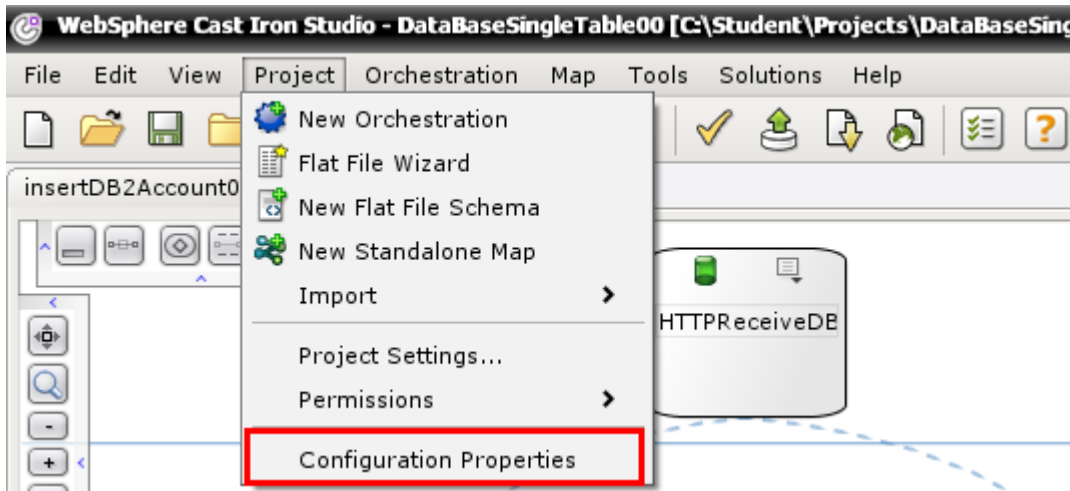
- __d. Double-click the function Substring Before and add the 'matchString' value '&'.
- __e. Right-click the graph pane and Apply Function Graph.

__12. Once all of the graphs have been applied the map should look like this:



__13. Create configuration parameters for the database connection parameters.

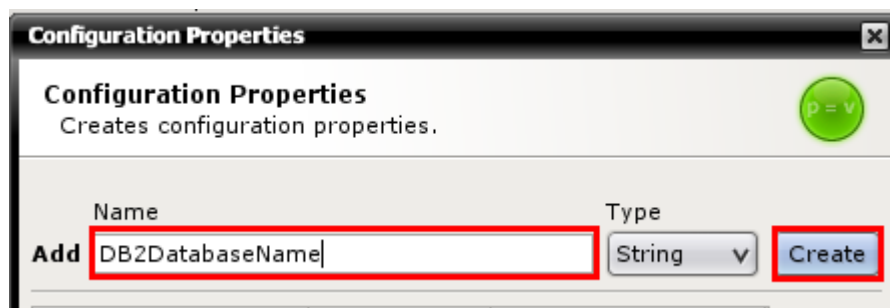
__a. From the Toolbar, go to *Project* → *Configuration Properties*.



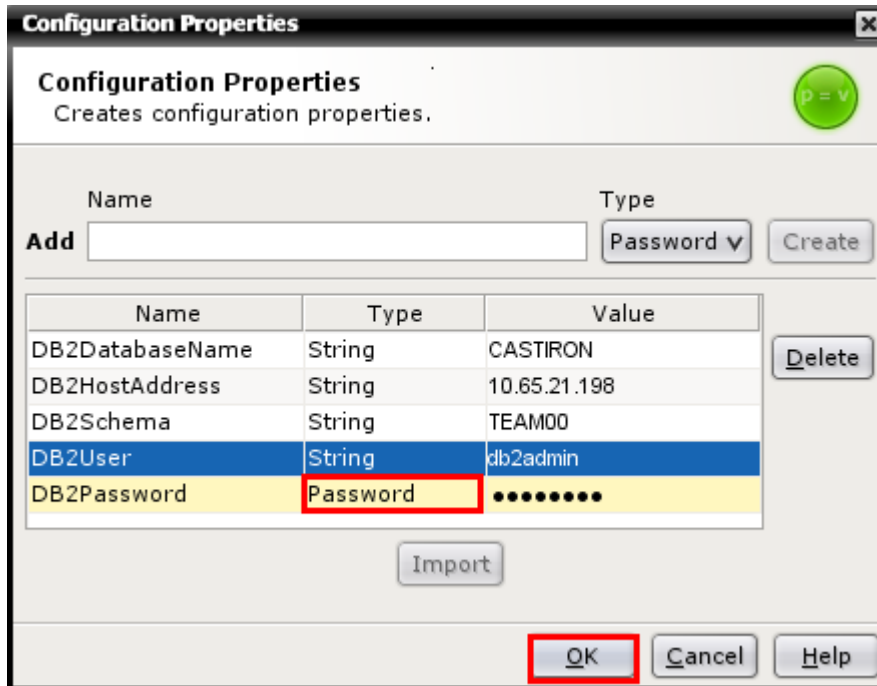
__b. Add each of the following parameters using values supplied in the '**Connection Parameters Spreadsheet**'.

1. DB2DatabaseName CASTIRON
2. DB2HostAddress XX.XX.XX.XX - Endpoint Server Address (DB2)
3. DB2Schema TeamXX
4. DB2User db2admin
5. DB2Password db2admin

***(type = password) see highlighted in **RED**.



- c. When all the parameters and values are added, select OK. Make sure you have added 'DB2Password' as the 'Password' Type.



- 14. Create a DB2 Database Endpoint.



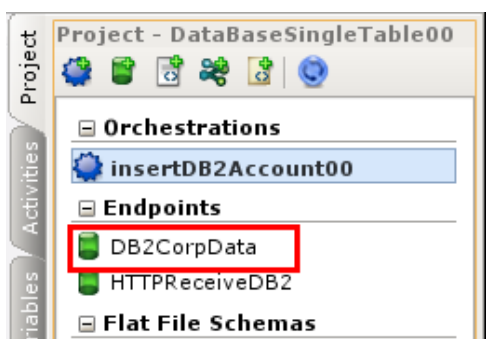
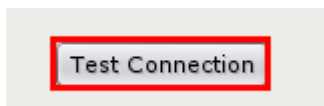
- a. From the '**Database Type**' drop-down, select '**DB2 UDB**'.
- b. Select the drop-down for the four fields corresponding to the configuration parameters entered in Step 4:

DB2DatabaseName, DB2HostAddress, DB2User, and DB2Password.

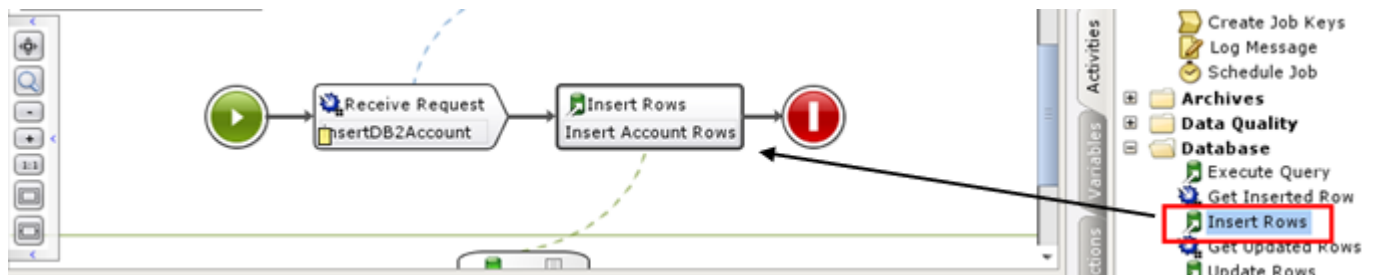
Under 'Additional Parameters' set PackageCollection to string NULLID.

Database Type	DB2 UDB	
Database Name	DB2DatabaseName	
Network Location	Server	Port
	DB2HostAddress	50000
Authentication	User Name	Password
	DB2User	DB2Password
Additional Parameters	Parameter Name	Parameter Value
	MaxPooledStatements	50
	PackageCollection	NULLID

- c. Save the project. At the bottom of the screen, click '**Test Connection**'. Once successful, then rename the endpoint to [DB2CorpData](#).



__15. Add a **Database Insert Rows** activity to the orchestration to update the Account table.



__a. **Summary:** Change the activity name to **Insert Account Rows**.

The screenshot shows the configuration window for the 'Database Insert Rows' activity, titled 'InsertAccountRows'. On the left is a 'Checklist' with the following items: 'Summary' (checked and highlighted with a red box), 'Pick Endpoint' (checked), 'Configure' (checked), 'Pick Table' (checked), and 'Delivery Rules' (checked). The main area is divided into two sections: 'Summary' and 'Activity Configuration Checklist'. The 'Summary' section has a text field for 'Activity Name' containing 'InsertAccountRows', which is highlighted with a red box. The 'Activity Configuration Checklist' shows a sequence of steps: 'Pick Endpoint (Select a database endpoint)' followed by 'Configure (Specify database cont)'. The 'Pick Endpoint' step is highlighted with a red box.

__b. **Pick Endpoint:** Browse for the **DB2CorpData** Endpoint.

The screenshot shows the configuration window for the 'Database Insert Rows' activity, titled 'InsertAccountRows', with the 'Pick Endpoint' tab selected. The 'Checklist' on the left has 'Pick Endpoint' highlighted with a red box. The main area shows the 'Pick Endpoint' configuration. The 'Database' field contains 'DB2CorpData' and is highlighted with a red box. To its right is a 'Browse...' button, also highlighted with a red box. Below these fields, the 'Database Type' is set to 'DB2 UDB' and the 'Database Name' is 'DB2DatabaseName'.

__c. **Pick table:**

- __i. Select the configuration parameter for the Schema field: TEAMXX
- __ii. Browse for the ACCOUNT table, and select OK.
- __iii. By default all fields are selected, but for this lab deselect 'Use All' and select the following columns (Reply to all of the warnings about null columns):
 1. ACCOUNT_NAME
 2. PHONE
 3. DESCRIPTION
 4. CREATEBYID
 5. LASTMODIFIEDBYID
 6. CREATEDATE
 7. LASTMODIFIEDDATE
 8. SYSTEMMODSTAMP

Checklist

- ✓ → Summary
- ✓ 📄 Pick Endpoint
- ✓ → Configure
- ✓ → **Pick Table**
- ✓ → Delivery Rules
- ✓ → Retry
- ✓ 🌐 Map Inputs
- ✓ 🌐 Map Outputs

Pick Table

Simple Database Operation

on Schema TEAM00 on Table ACCOUNT Browse...

Encoding ISO-8859-1

Select each column that will be affected by the database action.

Use All

Use	Column Name	Data Type	Encoding	Size	Nulla...
<input type="checkbox"/>	ID	BIGINT	ISO-8859-1	19	false
<input checked="" type="checkbox"/>	ACCOUNT_NAME	VARCHAR	ISO-8859-1	80	false
<input type="checkbox"/>	ACCOUNT_TYPE	VARCHAR	ISO-8859-1	40	true
<input type="checkbox"/>	PARENTID	VARCHAR	ISO-8859-1	18	true

__d. [Delivery Rules:](#)

__i. Select drop-down: 'At least once'.

__ii. Check: Get Row Counts.

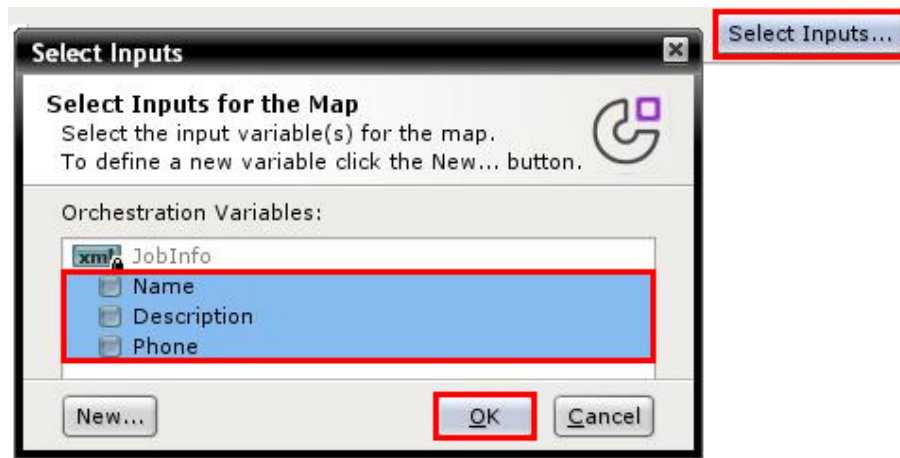


__e. [Map Inputs:](#)

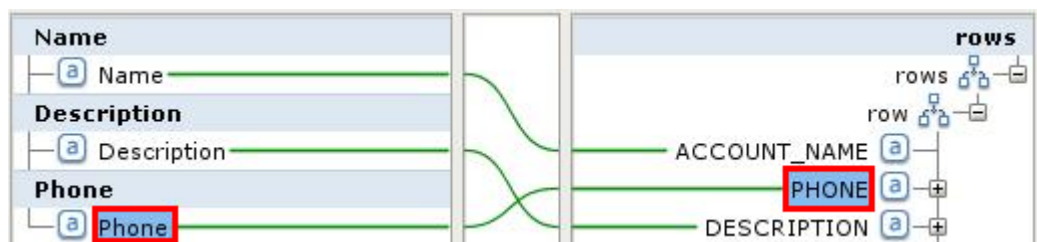
Note: Use this table for subsequent instructions:

DB2 Field	Orchestration Variable or function
ACCOUNT_NAME	Name
PHONE	Phone
DESCRIPTION	Description
CREATEDATE	Drag over Function: Get Current Date and Time
CREATEBYID	Enter Literal: Web
LASTMODIFIEDDATE	Drag over Function: Get Current Date and Time
LASTMODIFIEDBYID	Enter Literal: Web
SYSTEMMODSTAMP	Drag over Function: Get Current Date and Time

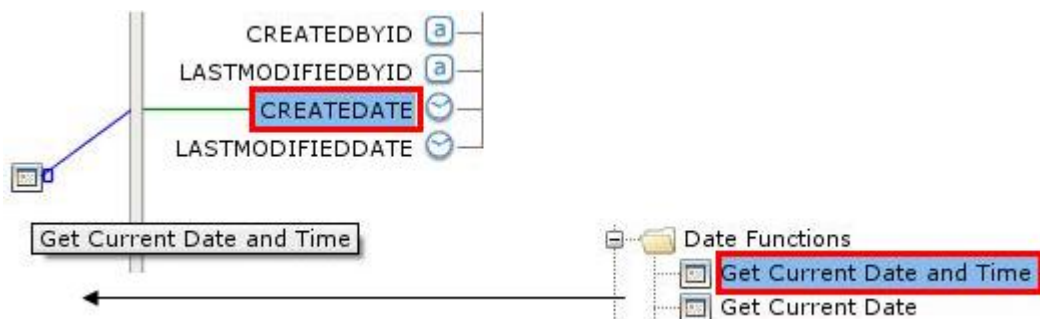
- __i. In the 'From Orchestration' pane, click the 'Select Inputs' button and select the three variables that were populated by the **HTTP Receive Request** activity.



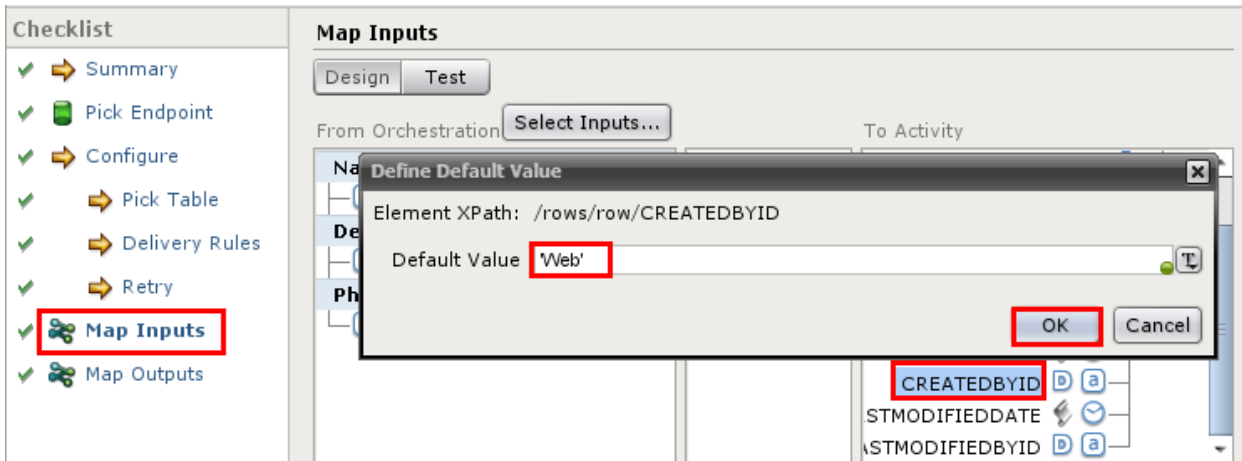
- __ii. Map the three input variables by dragging each input variable onto its corresponding Activity parameter.



- __iii. To apply the date function drag it **into the function graph pane**, map it to the CREATEDATE parameter, then right-click the mapping pane to Apply Function Graph. Repeat for LASTMODIFIEDDATE.



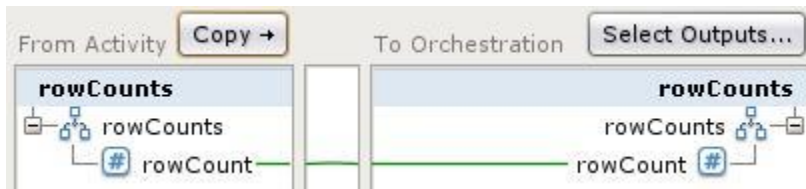
__iv. For the two id fields, right-click for option Define Default value, enter 'Web'.



__f. [Map Outputs:](#)

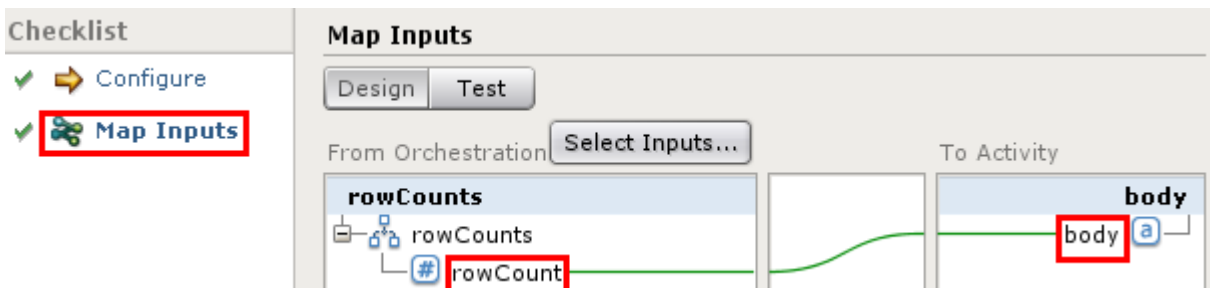
__i. The **Database Insert Rows** activity returns a count of rows inserted.

(1) Click the Copy button and select the **rowCounts** variable.



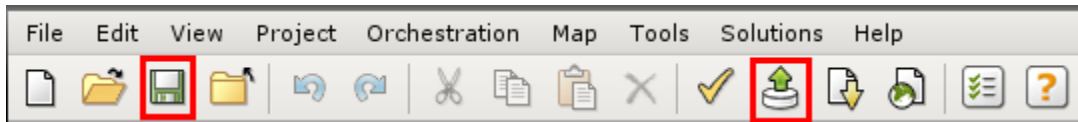
__16. Add an **HTTP Send Response** activity to the orchestration. Highlight the activity on the workspace.

__a. [Map Inputs:](#)



__i. To display the row count in the response, select the variable populated in the previous step and map it to **body**.

- __17. Save and Publish the project from the Toolbar to the appliance. Go to the WMC and deploy the project.



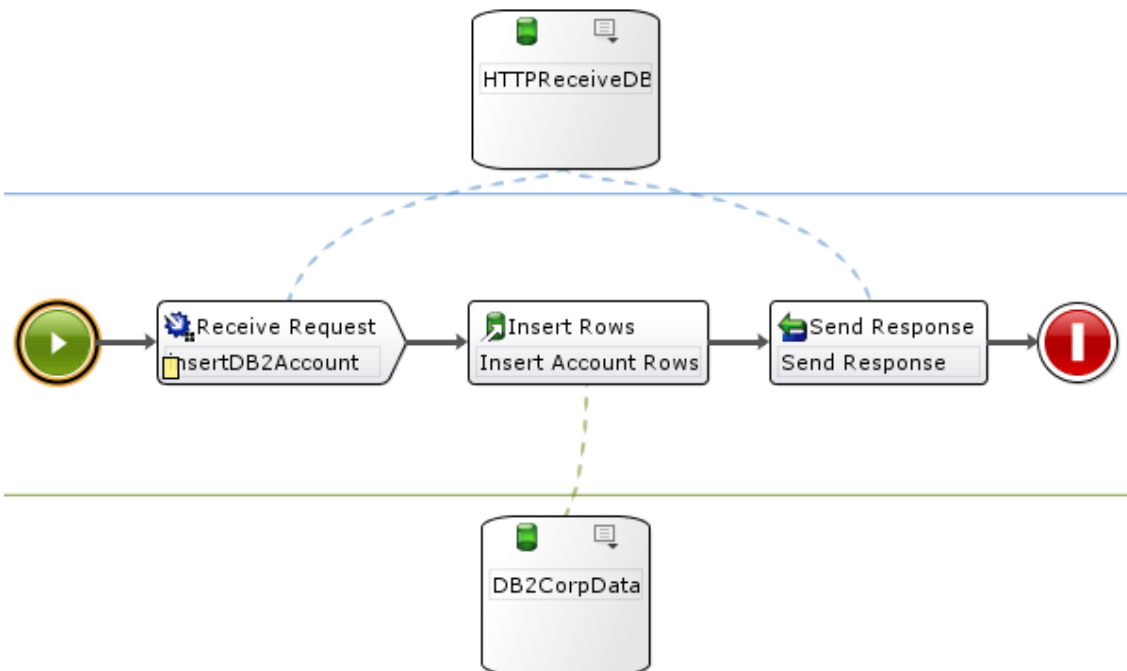
- __18. Test it using the 'accountForm.htm' file in the **C:\Student\Lab Resources\Lab 2** folder.
 - __a. Use the < Appliance EData Address (URL) > ip address from the '**Connection Parameters Spreadsheet**' and your custom URL.
 - __b. Fill in the input fields as needed and Submit.

A screenshot of a web form titled "Create DB2 Account" from CAST IRON SYSTEMS. The form contains the following fields and values:

URL of inbound connector:	http://174.168.36.127/insertDB2Account99
Name:	Enterprise Software Resale
Description:	Reseller of pirated software
Phone:	888-920-1110

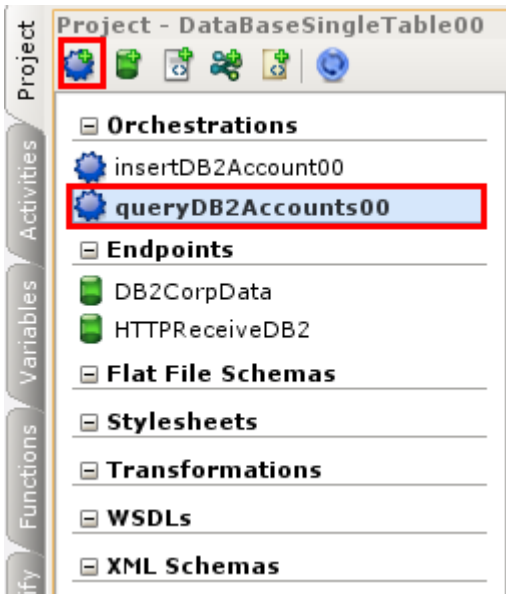
At the bottom of the form are two buttons: "Submit" and "Reset".

- __c. A single '1' will be returned as the row count if the insertion was successful.



2.6 Return All Accounts from Database

- __1. Create a new orchestration and rename it to `queryDB2AccountsXX`.

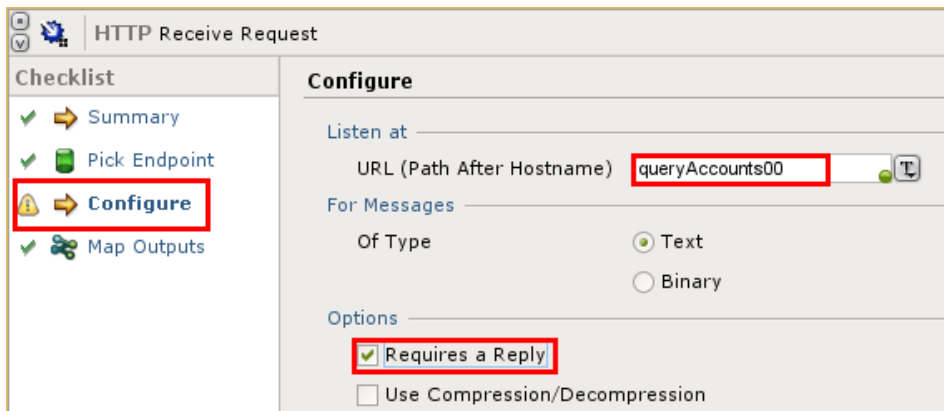


- __2. Add an **HTTP Receive Request** activity to the orchestration and rename it to `queryDB2Accounts`
 - __a. **Pick Endpoint:** Browse for the HTTP Endpoint.

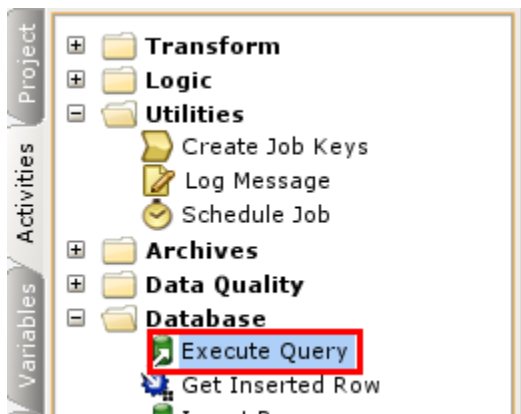


__b. [Configure](#):

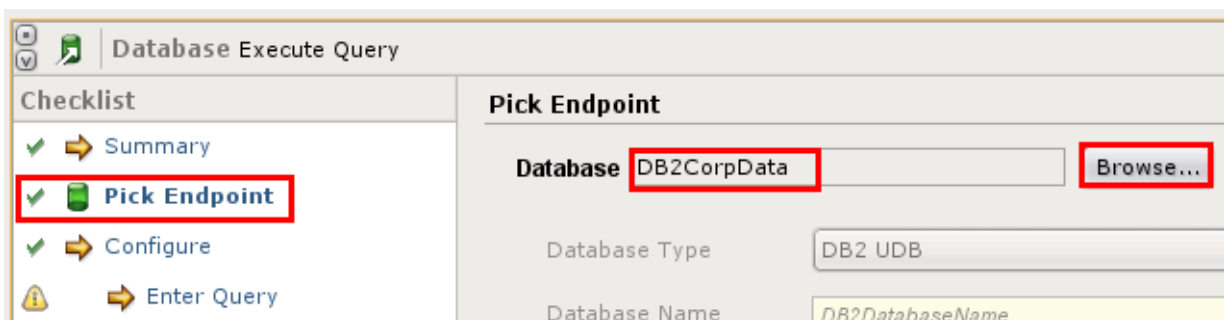
__i. URL = "queryAccountsXX". Select option 'Requires a Reply'.



__3. Add a **Database Execute Query** activity to the orchestration and rename it to [AccountQuery](#).

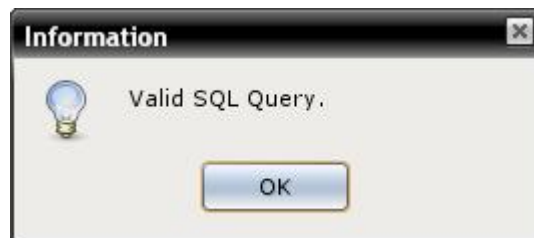
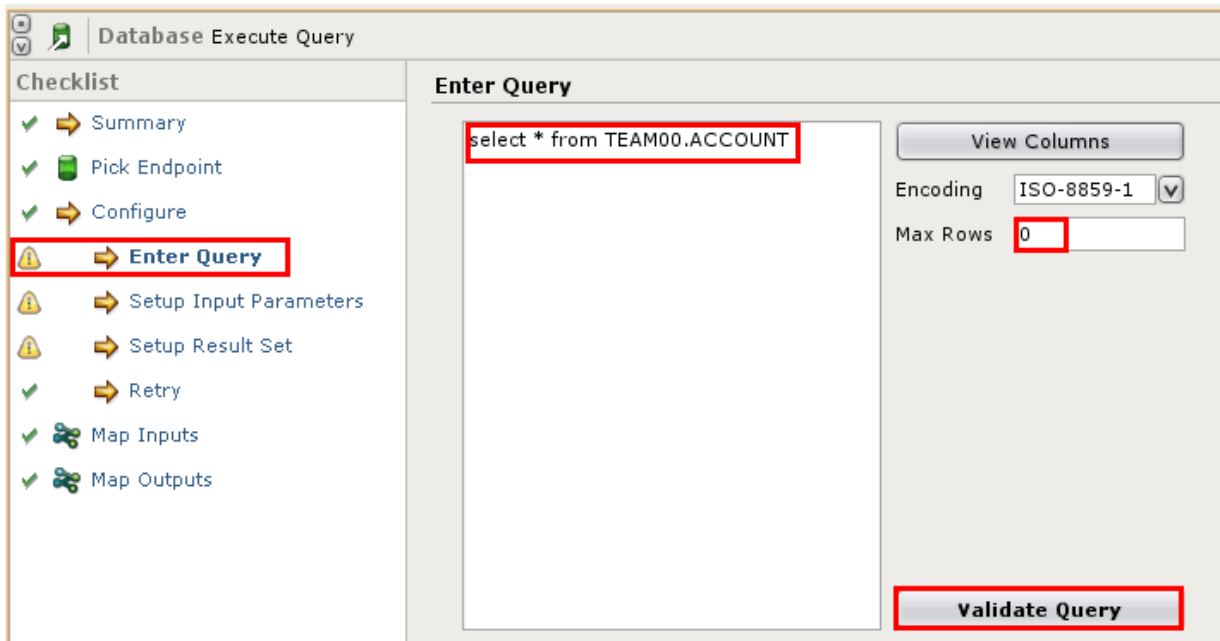


__a. [Pick Endpoint](#): Browse for the DB2 Endpoint.



__b. **Enter Query:**

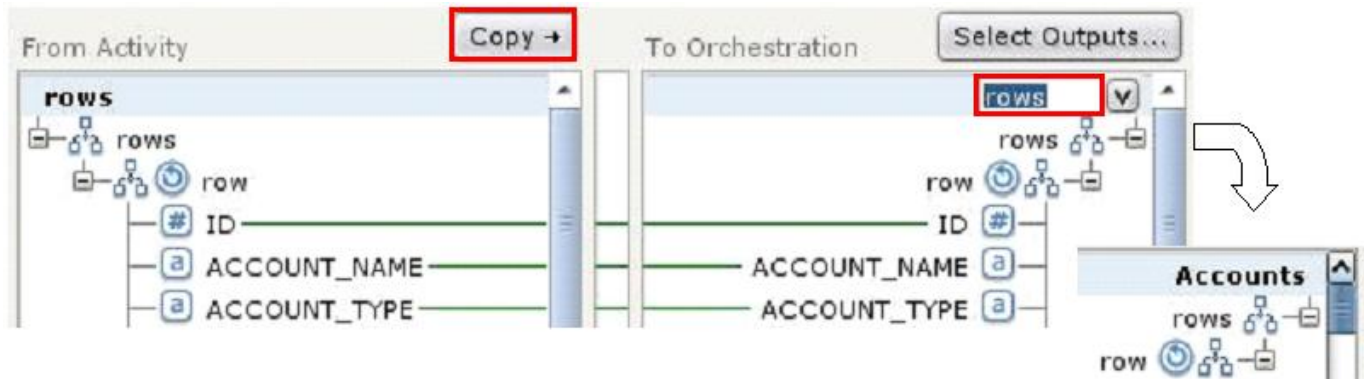
- __i. Find your schema in the '**Connection Parameters Spreadsheet**' and enter `select * from TEAMxx.ACCOUNT` in the **Enter Query** field.
- __ii. When Max Rows is set to 0, all matching rows will be returned.
- __iii. Click the **Validate Query** button to check the syntax of the query and the connection to the database.



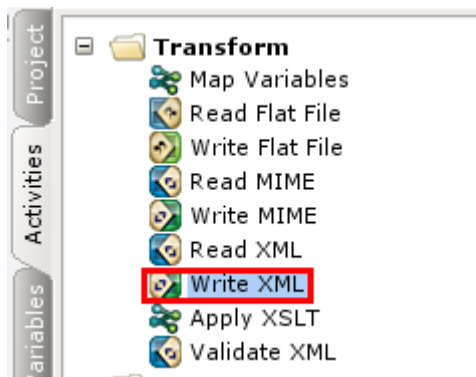
__c. **Map Outputs:**

__i. Copy the output of the activity to the orchestration:

- (1) Click the **Copy** button and select the activity's output parameter.
- (2) Rename the new orchestration variable to **Accounts**.



__4. Add a **Transform Write XML** activity to the orchestration.

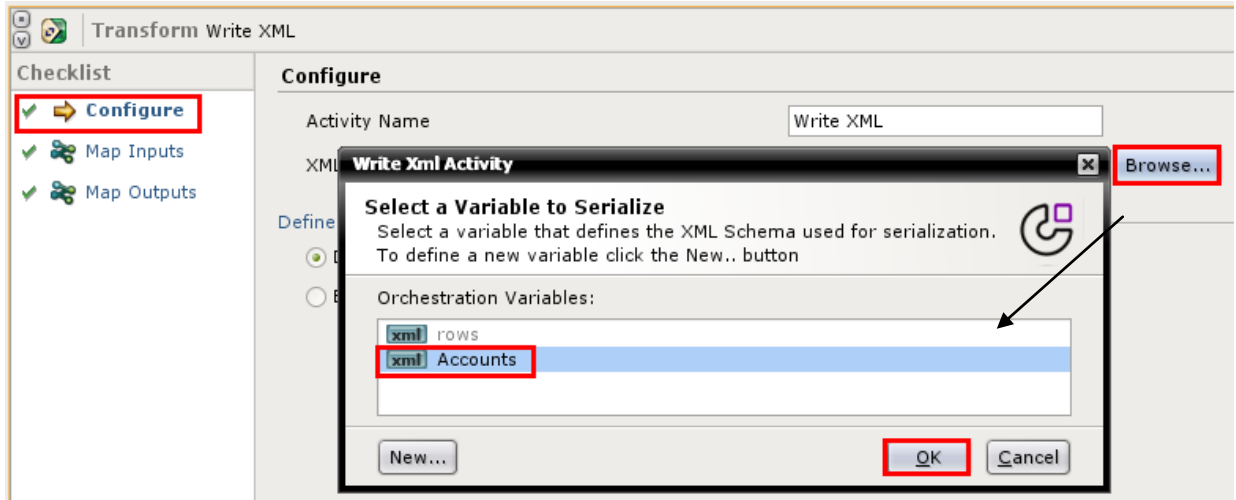


__a. This activity converts the XML produced by the database query to a string that can be used in the HTTP response. Given a definition of a data type defined in an XML schema, this activity will convert it to a string.

b. Configure

i. Click the Browse button to find the XML schema to be used for parsing.

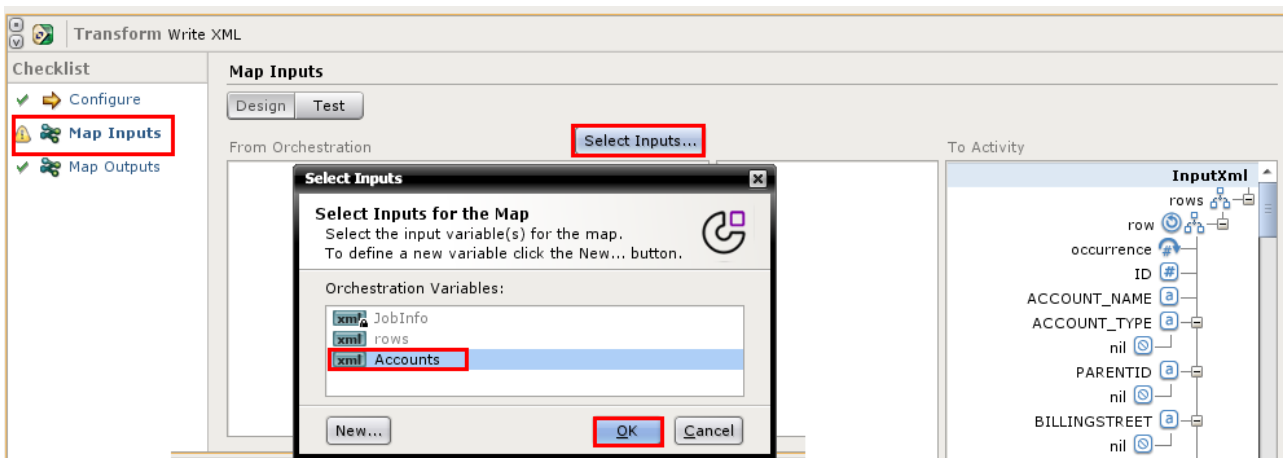
This schema will be derived from the **Accounts** variable that was created in the previous step by the **Execute Query**.



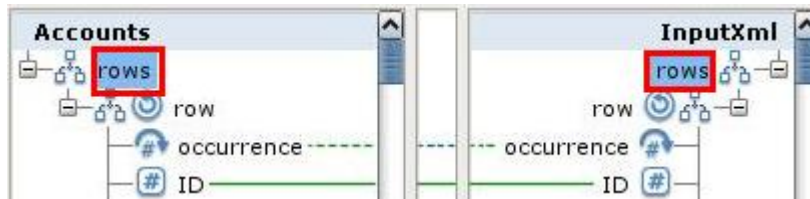
c. Map Inputs:

i. Use the Select Inputs button to load the **Accounts** variable populated by the query activity.

Notice that the Activity mapping pane already has the Accounts structure due to the schema specification made in the **Configure** step.

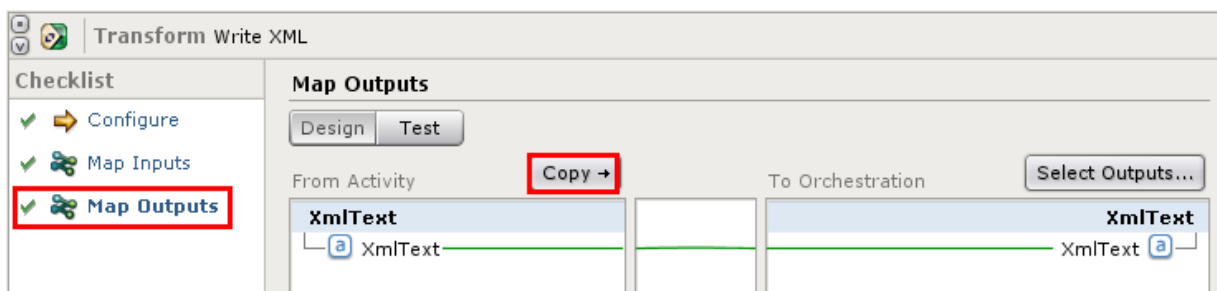


__ii. Map the top-level **rows** orchestration variable to the activity variable.



__d. **Map Outputs:**

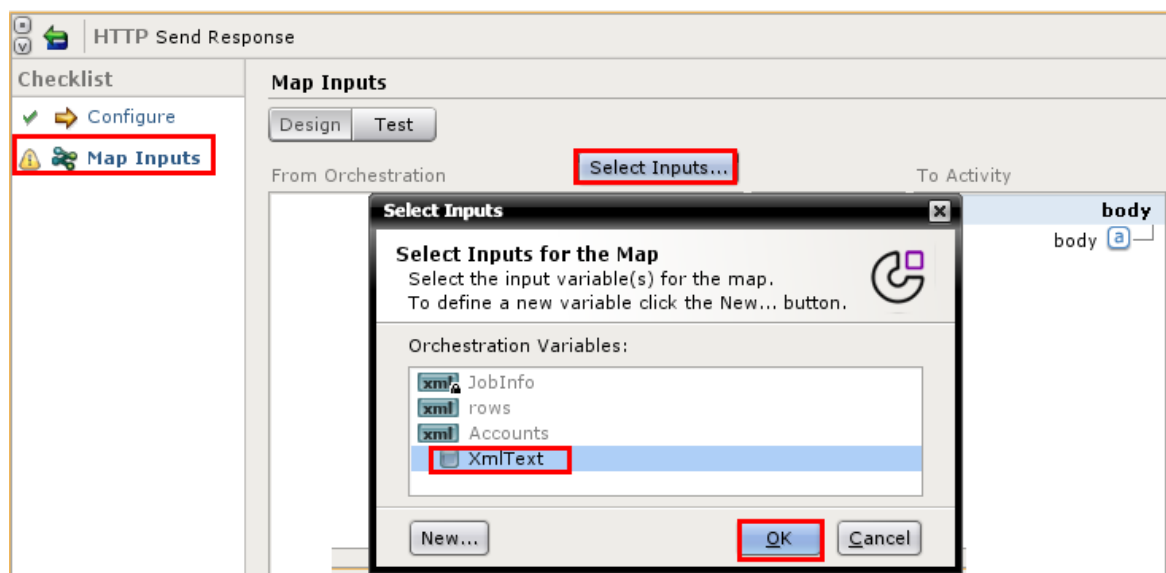
__i. Use the **Copy** button to copy the transformed activity output to the orchestration.



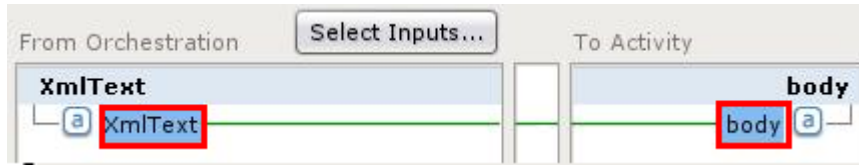
__5. Add an **HTTP Send Response** activity to the orchestration.

__e. **Map Inputs:**

__i. Use the Select Inputs button to load the XmlText orchestration variable created by the previous Write XML activity.



- __ii. Map the **XmlText** variable to the **body** of the response.



- __6. Save and Publish the project to the appliance, update the configuration parameters and deploy it.
- __7. Test with the HTTP Post Utility.
 - __a. Use the < Appliance EData Address (URL) > IP address from the '**Connection Parameters Spreadsheet**' and your custom URL.



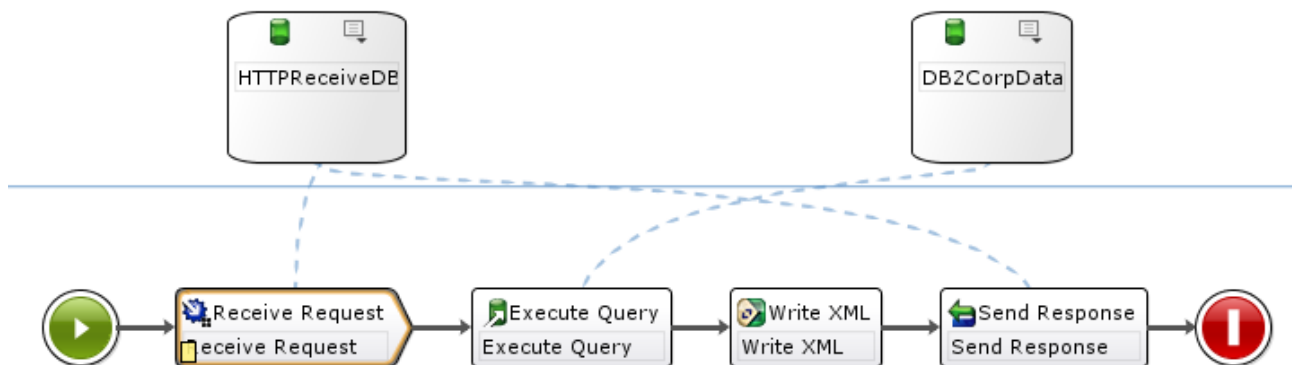
CAST IRON

Post a document to the WebSphere Cast Iron Integration Orchestration

URL of inbound connector:

Document to post:

```
<?xml version="1.0" encoding="UTF-8" ?>
- <rows xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
- <row>
  <ID>100</ID>
  <ACCOUNT_NAME>IBM</ACCOUNT_NAME>
  <ACCOUNT_TYPE>(null)</ACCOUNT_TYPE>
  <PARENTID>(null)</PARENTID>
```



Appendix A. Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have

been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental. All references to fictitious companies or individuals are used for illustration purposes only.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Appendix B. Trademarks and copyrights

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

IBM	AIX	CICS	ClearCase	ClearQuest	Cloudscape
Cube Views	DB2	developerWorks	DRDA	IMS	IMS/ESA
Informix	Lotus	Lotus Workflow	MQSeries	OmniFind	
Rational	Redbooks	Red Brick	RequisitePro	System i	
<i>System z</i>	<i>Tivoli</i>	<i>WebSphere</i>	<i>Workplace</i>	<i>System p</i>	

Adobe, Acrobat, Portable Document Format (PDF), and PostScript are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, other countries, or both.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both. See Java Guidelines

Microsoft, Windows, Windows NT, and the Windows logo are registered trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

ITIL is a registered trademark and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

Other company, product and service names may be trademarks or service marks of others.



© Copyright IBM Corporation 2011.

The information contained in these materials is provided for informational purposes only, and is provided AS IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, these materials. Nothing contained in these materials is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software. References in these materials to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. This information is based on current IBM product plans and strategy, which are subject to change by IBM without notice. Product release dates and/or capabilities referenced in these materials may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way.

IBM, the IBM logo and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at ibm.com/legal/copytrade.shtml

Other company, product and service names may be trademarks or service marks of others.



Please Recycle
