

IBM Integration Bus

Implementing Analytics with the R Node using an Integration Service

Featuring:

Analytics using R Integration service

June 2015 Hands-on lab built at product Version 10.0.0.0

Contents

1. INT	RODUCTION	3
2. PR	EPARE THE ENVIRONMENT	4
2.1 2.2 2.3 2.4	INSTALL THE IIB R NODE OPEN THE WINDOWS LOG MONITOR FOR IIB START R SERVER IMPORT PRE-REQUISITES	
3. UP	DATE EMPLOYEE SERVICE APPLICATION TO CALL R	10
3.1 3.2 3.2 3.2 3.3	ADD OPERATION TO EMPLOYEESERVICE CONFIGURE AN R NODE	
4. TE	STING	
4.1 4.2 4.3 4.4	DEPLOY THE SERVICE TEST THE SERVICE WITH THE IIB FLOW EXERCISER UNDERSTANDING THE VALUE RETURNED FROM R ANALYZE THE MODEL BEFORE AND AFTER RUNNING THE APPLICATION	
		38

1. Introduction

This lab guide provides instructions on how to set up an R node in IIB and use it in an integration service. Data was extracted from the EMPLOYEE table in the DB2 SAMPLE database and an R model was created with the data. The simple application that you will create, will predict the value of Salary given a value for Education level. The following diagram outlines at a high level what you will do:



This lab guide shows you how to do the following tasks:

- 1) How to add R node support to your environment using the R node IIB Extension.
- 2) How to configure a simple message flow containing an R node using an integration service.
- 3) How to configure an R node in a message flow.
- 4) How to test a message flow with an R node and interface with R serve.
- 5) How to understand the number returned from R.

2. Prepare the Environment

2.1 Install the IIB R node

The Analytics node "R" is not part of the standard IIB V10 toolkit and runtime installation. The R node is being made available as an IIB extension from GitHub and is available from here:

https://github.com/ot4i/integrate-R.

In order to do this lab, you will need to install the R node. Two files need to be copied into the IBM Integration Bus installation. They are in C: $\studentl0\Analytics\Resources$ and are:

- A toolkit component (currently RNodeToolkit_1.0.0.20150417-1641.jar): to be copied to C:\IBM\IIB\10.0.0.0\tools\plugins. (Once this has happened, the Integration Toolkit must be restarted).
- A runtime component (currently RNodeRuntime-1.0.0.20150417-1641.par): to be copied to C:\IBM\IIB\10.0.0\server\bin. (Once this has happened, any integration servers to which the R node will be deployed, must be restarted).

You can do this manually, if you wish. Alternatively, run a script we written for you, as follows:

- 1. Open an IIB Console and navigate to "C:\student10\Analytics\Commands" and run the command RnodeInstall.cmd
- The script defaults to copying the files named above from C:\student10\Analytics\Resources to their respective destinations. In addition, it stops and starts IB10NODE by default. Accept these defaults by pressing Enter.

3. The output from the script should be like this:

```
C:\student10\Analytics\Commands>RnodeInstall.cmd
This command file must be run within an Integration Bus Command Console.
Betaworks Analytics Lab guide Rnode setup.
Enter IIB Node name (default is IB10NODE):
Enter R Node runtime file name (default is RNodeRuntime-1.0.0.20150417-1641.par)
Enter R Node runtime from folder (default is C:\student10\Analytics\Resources):
Enter R Node runtime to folder (default is C:\IBM\IIB\10.0.0.0\server\bin):
Enter R Node toolkit file name (default is RNodeToolkit 1.0.0.20150417-1641.jar)
Enter R Node toolkit from folder (default is C:\student10\Analytics\Resources):
Enter R Node toolkit to folder (default is C:\IBM\IIB\10.0.0.0\tools\plugins):
Thankyou, using values "IB10NODE", "RNodeRuntime-1.0.0.20150417-1641.par", "C:\s tudent10\Analytics\Resources", "C:\IBM\IIB\10.0.0.0\server\bin", "RNodeToolkit_1
.0.0.20150417-1641.jar", "C:\student10\Analytics\Resources", "C:\IBM\IIB\10.0.0.
0\tools\plugins"
Ok to proceed? Use Ctrl-C to terminate.
Press any key to continue . . .
Stopping node "IB10NODE"
BIP8019E: Integration node 'IB10NODE' stopped.
This integration node is stopped; the command you issued cannot be processed whe
n an integration node is stopped.
A previous command has been issued to stop this integration node, or this integr
ation node has never been started.
This integration node can be started, changed, or deleted.
About to remove R Node code from IIB. Failure messages can be ignored.
Could Not Find C:\IBM\IIB\10.0.0.0\server\bin\RNodeRuntime-1.0.0.20150417-1641.p
ar
Could Not Find C:\IBM\IIB\10.0.0.0\tools\plugins\RNodeToolkit_1.0.0.20150417-164
1.jar
About to copy across R Node code.
Ok to proceed? Use Ctrl-C to terminate.
        1 file(s) copied.
        1 file(s) copied.
About to Start "IB10NODE"
BIP8096I: Successful command initiation, check the system log to ensure that the
 component started without problem and that it continues to run without problem.
```

A Check that the files have been copied across and the integration node has been started.

2.2 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.

🔝 IIB Event Log Monitor	
BIP31321: < IB10NODE.server1 > The HTTP Listener has started listening on port	' 🔺
77800'' for 'http' connections. [10/3/2014 3:17:23 PM]	
DIF21341: (IBLUMUDE.Server1) Execution group finished with configuration mes-	sa
90. [10/3/2014 3.17.24 [1] DIP21521: (ID10NODE convent) Configuration massage received from broken [10	12
2014 5:07:36 PM1	
[10/3/2] BIP21531: (IBI0NODE server1.) About to 'Change' an execution group. [10/3/2]	91
4 5:07:36 PM]	
BIP2155I: (IB10NODE.server1) About to ''create '' the deployed resource ''Em	pl
oyeeService_JSONClient'' of type ''.APPZIP''. [10/3/2014 5:07:37 PM]	
BIP2155I: < IB10NODE.server1 > About to ''create '' the deployed resource ''ge	n.
getEmployee_EmployeeService_EmpServClient_JSON1'' of type ''.SUBFLOW''. [10/3/	20
14 5:07:37 PM1	
BIP21551: (IB10NODE.server1) About to ''create '' the deployed resource ''Em	pS
ervClient_JSON1' of type ''.MSGFLOW''. 110/3/2014 5:07:37 PMJ	
BIP21541: (IBIØNUDE.server1) Execution group finished with Configuration mes	sa
ge. $[10/3/2014 5:07:43 PM]$	
BIP31321: (IBIONUDE.HIIPLISTENER) INE HIIP LISTENER has started listening on	p
ort 7080' for http://connections. L10/3/2014 5:07:47 Fmj	12
BIT21521: (IBIDNUDE.SERVERI) Configuration message received from proker. [10	⁻³
72014 3-30-41 ml	74
A E-CA-A4 DM1	21
T 3-30-11 [1] BIP215I: (] BIGNODE sequent) Obout to 22delete 22 the deployed vecouves 22Fm	
Difficulture in the second of	po l

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

2.3 Start R server

In order to run a message flow with an R node, the R server needs to be started. In this section you will start R Server from the R GUI.

1. From the Windows Start menu start the R Gui, (click Start > All programs > R > "R x64 3.1.2")

🕌 IBM WebSphere	I
IBM WebSphere AppServer	L
📗 IBM WebSphere MQ	L
li Maintenance	L
Microsoft Visual Studio 2012	L
Microsoft Visual Studio 2012 Express	L
📗 Mozilla Firefox	L
📔 Node.js	L
Notepad++	L
📙 R	L
R i386 3.1.2	L
🥂 R x64 3.1.2	L
SmartBear	L
🔒 Startup	L
Symantec Endpoint Protection	1
🔒 Utilities	
📔 Windows Kits 📃	1
✓ Back	
Search programs and files	
Start 🛛 🏹 🎱 🕖 Administrator: IB	м

2. The RGui(64-bit) with an R Console will open:



3. In the "R console" type:

library(Rserve)

and press enter.

4. In the same console type:

Rserve()

and press enter.

The R Console will look like this:

RGui (64-bit)	
File Edit View Misc Packages Windows Help	Į
R Console	- U X
Type 'q()' to quit R.	-
> library(Rserve)	
> Rserve()	
Starting Rserve	
"C:\Users\iibadmin\DOCUME~1\R\WIN-LI~1\3.1\Rserve\libs\x64\Rserve.exe	:"
	- I

The IIB R node can now use Rserve to interface with R.

Page 8 of 38 Implementing Analytics with the R Node using an Integration Service Version 10.0.0.0

2.4 Import pre-requisites

- 1. First, to ensure there are no conflicts with other components, switch to a new IIB workspace in the IIB Toolkit, for example name it c:\workspaces\IIB_R.
- 2. In the IIB Toolkit, import these Project Interchange files: c:\student10\Integration_service\solution\EmployeeServiceInterface.V10.zip and c:\student10\Integration_service\solution\ EmployeeService.V10.zip

3. Update Employee Service application to call R

3.1 Add Operation to EmployeeService

1. Go to the EmployeeService integration service and double-click the Integration Service Description in the navigator.

Select the Interface tab. You will see operations, getEmployee and updEmployee.

SM EmployeeServ	vice 🛛				
▼Interface					
Configuration					
Name	EmployeeServio	imployeeService			
Namespace	http://Employee	eService			
Location	/EmployeeServi	ceInterface/EmployeeService.wsdl			
•Operations an	s 🐉 🐉	• 🖻 🖓 🦨 😹 💬			
Message Typ	be	Name	Туре		
🔻 👹 getEm	ployee				
Di getEmplo	oyee	EMPLOYEE	EMPLOYEE		
getEmplo	oyeeResponse	EmployeeResponse	EmployeeResponse		
🔻 👹 updEm	nployee				
D updEmpl	oyee	EMPLOYEE	EMPLOYEE		
() updEmpl	oyeeResponse	EmployeeResponse	EmployeeResponse		
5 Service	Interface				

2. Click the Add Request Response Operation button which will create another operation called "operation1".

Overtype this to name the new operation "predictSalary".

Press return, which will automatically change the names of the input and output message types (predictSalary and predictSalaryResponse).

🗹 *EmployeeService 🛛		
▶ Interface		
Configuration		
▼Operations Operations and their parameter) 🖻 🔊 🔊 😹 📰	
Message Type	Name	Туре
🔝 🎲 getEmployee		
D getEmployee	EMPLOYEE	EMPLOYEE
getEmployeeResponse	EmployeeResponse	EmployeeResponse
🔻 👹 updEmployee		
Di updEmployee	EMPLOYEE	EMPLOYEE
(1) apdEmployceResponse	EmployeeResponse	EmployeeResponse
▼ predictSalary		
D] predictSalary	input1	string
I predictSalaryResponse	output1	string

3. Click the word "string" to change the type of both the input and output messages to EMPLOYEE (use the Browse button when the dialogue window opens, and type "e" to show all available types that beginning with "e".

Save the changes to the service.

🗺 *EmployeeService 🛛				
▶Interface				
Configuration				
▼Operations) 🖻 🖓 🎝 💹 🚟			
Operations and their parameter	ers			
Message Type	Name	Туре		
🔝 🎆 getEmployee				
D getEmployee	DigetEmployee EMPLOYEE EMPLOYEE			
getEmployeeResponse	EmployeeResponse	EmployeeResponse		
🔻 👹 updEmployee				
D updEmployee	EMPLOYEE	EMPLOYEE		
I updEmployeeResponse EmployeeResponse EmployeeResponse		EmployeeResponse		
▼ [*] ∰ predictSalary				
D predictSalary	Di predictSalary EMPLOYEE EMPLOYEE			
I predictSalaryResponse	I predictSalaryResponse EMPLOYEE EMPLOYEE			

4. Click the Service tab to show the implementation. Note that predictSalary is greyed out, since it has not yet been implemented.

🗺 EmployeeService 🛛	
S EmployeeService >	
S EmployeeService	EmployeeService
SOAP/HTTP Binding	igetEmployee
	👹 updEmployee
	aredictSalary
	🔀 Error Handlers
	🗉 Eailure
	□ _□ <u>Catch</u>
	🗐 Timeout
🔄 Service 🕕 Interface	

5. Click predictSalary, which will open the subflow editor.

3.2 Configure an R Node

3.2.1 Configure the Basic properties

1. Open the Analytics folder in the message flow palette, and drag the R node onto the canvas:



2. Highlight the R node so that the properties tab shows the R node properties:

🔲 Properties 🛛 🔓	🛚 Problems 🗦 Out	ine 🧟 Tasks 🖽 Deployment Log	2	
🕼 R Node Prop	erties - R			
Description	🛞 Rserve server	: A value must be set for this property.		
Basic	Rserve server*	<rserve a="" by="" colon="" optional="" port,="" separated="" server="" with=""></rserve>		
Variables	RData file	<rdata a="" an="" connection="" established="" file="" is="" load="" rserve="" server="" to="" when=""></rdata>		
Connection Pool Monitoring	Connect script <pre> <r a="" an="" connection="" established="" execute="" is="" rserve="" script="" server="" to="" when=""></r></pre>			
	Evaluate script*	<r a="" by="" execute="" is="" message="" node="" processed="" script="" this="" to="" when=""></r>		
	Disconnect script	<r a="" an="" before="" connection="" dosed="" execute="" is="" rserve="" script="" server="" to=""></r>		

3. Configure the R node properties as follows:

Basic Tab:

Rserve server : localhost: 6311

RData file: C:\student10\Analytics\RWork\employeedata.RData

(note: in a distributed environment - the RData file needs to exist on the same machine as IIB toolkit)

Connect script: C:\student10\Analytics\RWork\RnodeInit.r Evaluate script: C:\student10\Analytics\RWork\RnodeEvaluate.r Disconnect script: C:\student10\Analytics\RWork\RnodeDisconnect.r

(note: in a distributed environment- the scripts need to exist on the same machine as IIB Run time environment, they also reference locations on the server where R is installed)

🔲 Properties 🖾 🚺	🖞 Problems 🗄 Out	ine 🖉 Tasks 🖽 Deployment Log	2	° 🗆
🧟 R Node Prope	erties - R			
Description				
Basic	Rserve server*	localhost:6311		
Variables	RData file	C:\student10\Analytics\RWork\employeedata.RData		
Connection Pool Monitoring	Connect script	C:\student10\Analytics\RWork\RnodeInit.r		
	Disconnect script	C:\student10\Analytics\RWork\RnodeDisconnect.r		

3.2.2 Configure Variables

The Variables tab in the R node is where you configure the variables to be used with your R environment. A Data Frame, in R, is a used to store a collection of data in rows and columns, (similar to the concept of a matrix in mathematics, however the types of a data frame do not necessarily need to be numeric).

In the following section you will define the variables that will be used in R and how IIB will use them.

1. Variables Tab:

Click the "Add..." button to add a variable, the "Add Variable bindings entry" window will appear: Specify the following (case sensitive):

R variable name: **newdata** R variable type : **Data Frame** Direction : **In**

For Data location click the Edit button:

Add Variable bindings entry
R data frame name
R variable name*
newdata
R variable type
Data Frame
Direction
In
Data iocation*
°I
Oata location: A value must be set for this property.
OK Cancel

2. The XPath Expression Builder will open.

In the Data Types Viewer section, expand \$Root and click "(Add Data Type)"

XPath Expression Builder	
(Path Expression Builder	
Select the target from the Schema viewer, Function viewe below.	r or Operator viewer and dra
Data Types Viewer	XPath Functions
Image: Constraint of the second state of the second st	Dolean Dolean Dolean DoleSet DoleSet
Show XML Schema groups XPath Expression	

- 3. In the "Type Selection" window,
 - 1) highlight **EMPLOYEE**
 - 2) Change the Domain to **XMLNSC**
 - 3) and click **OK**:

Type Selection	
Choose a type (? = any character, * = any s	string):
, Matching types:	
(a) actor	
(a) base	
e Body	
e DBResp	
E EMPLOYEE	
(a) encodingStyle	
e Envelope	
e Fault	
e getEmployee	
e getEmployeeResponse	
e Header	-
Qualifier:	_
ttp://sample/iibadmin	
Domain: XMLNSC	-
ОК	Cancel

4. The EMPLOYEE data type appears under \$Root.

Double click on **sam:EMPLOYEE:EMPLOYEEType [XMLNSC]** to set the value in the XPath **Expression**

Click Finish to save the expression:

🕟 XPath Expression Builder			<u>- 🗆 ×</u>
XPath Expression Builder Select the target from the Schema viewer, Function viewer or Ope below.	erator viewer and drag and drop the	nodes in the source viewer	55
Data Types Viewer	XPath Functions	Operators	
Data Types	E String		_
(Add Data Type)		· · · //	
e sam:EMPLOYEE : EMPLOYEEType [XMLNSC]	🗄 🖷 NodeSet	<=	
H. C MQMD : MQMD_TYPE	🗄 🖄 Axes		
Properties : PropertiesType		8dt;	
. IMSTransport : JMSTransport_type			
			-
	,	, , .	
XPath Expression			A V
Namespace settings			
?		Finish	Cancel

5. The Data location field will be automatically completed with the XPath expression.

Click OK to add the variable:

Edit Variable bindings entry
R data frame name
R variable name*
newdata
R variable type
Data Frame
Direction
In
Data location*
\$Root/XMLNSC/sam:EMPLOYEE Edit
OK Cancel

Page 18 of 38

Implementing Analytics with the R Node using an Integration Service Version 10.0.0.0

6. The variable bindings table will be updated with the definition for the newdata "Data Frame":

🏼 Properties 🕱 🔝 Problems 🗄 Outline 🖉 Tasks 🖽 Deployment Log	
R Node Properties - R	
Description	
Basic Variable bindings R data frame name R variable name R variable type Direction Data location	Add
Variables newdata Data Frame In \$Root,XMLNSC/sam:EMPLOYEE	e dia
Connection Pool	Edit
Manitarian	Delete

7. Click the Add button to add a second variable:

Click the "Add..." button to add a **second variable**, the "Add Variable bindings entry" window will appear.

Specify the following (case sensitive):

R data frame name : **newdata** R variable name : **salary** R variable type : **Integer** Direction : **Out**

For Data location click the Edit button:

Add Variable bindings entry
R data frame name
newdata
R variable name*
salary
R variable type
Integer 💌
Direction
Out
Data location*
[Edit]
😣 Data location: A value must be set for this property.
OK Cancel

8. In the XPath Expression Builder expand \$Root until you can select SALARY. Double click on Salary to formulate the expression. Click Finish:

ND VDath Evoraccion Builder			
VD-th Expression Builder			
Select the target from the Scheme viewer. Exection viewer or Op	erator viewer and drag an	d drop the	$\langle \langle \cdot \rangle$
nodes in the source viewer below.		iu urop ure	\sim
Data Types Viewer	XPath Functions	Operators	
E- B Data Types	E String		-
⊡KP= \$Root (Add Data Type)	Boolean		
are (dot bata type) □e sam:EMPLOYEE : EMPLOYEEType [XMLNSC]	Die Set		
EMPNO : string	🗄 🖄 Axes		
P FIRSTNME : string		<=	
LASTNAME : string		··· >=	
WORKDEPT : string			
PHONENO : string		!=	
e JOB : string		and	
EDLEVEL : short	1	+	
SEX : string		-	
C SALARY : decimal		div	
BONUS - decimal		mod	
COMM : decimal	1	Г., "	-
Show XML Schema groups	.,		
YPath Expression			
\$Root/XMLNSC/sam:EMPLOYEE/SALARY			<u> </u>
			-
Namespace settings			
0		nish C	ancel
\odot			

9. The Data location will automatically be filled with the location of SALARY in the message tree, Press OK to save the variable:

ſ	Add Variable bindings entry
	R data frame name
	newdata
	R variable name*
	salary
	R variable type
	Integer 💌
	Direction
	Out 💌
(Data location*
l	\$Root/XMLNSC/sam:EMPLOYEE/SALARY
	OK Cancel

10. Click the "Add..." button to add a **third variable**, using the same method you used for the previous two variables.

Specify the following (case sensitive):

R data frame name:newdataR variable name:R variable type:Direction:InData location:\$Root/XMLNSC/sam:EMPLOYEE/EDLEVEL

Add Variable bindings entry
R data frame name
newdata
R variable name*
ed.level
R variable type
Integer 💌
Direction
In
Data location*
v \$Root/XMLNSC/sam:EMPLOYEE/EDLEVEL Edit
OK Cancel

Page 21 of 38

11. The variables table will look like the following when all required variables for the model we are using are defined:

🔲 Properties 🛛 🚼 Problems 🗄 Ou	utline	🖉 Tasks 🔠 Deploym	ent Log			
R Node Properties - R						
Description						
Basic Variable bindings		R data frame name	R variable name	R variable type	Direction	Data location
Variables			newdata	Data Frame	In	<pre>\$Root/XMLNSC/sam:EMPLOYEE</pre>
Connection Pool		newdata	salary	Integer	Out	\$Root/XMLNSC/sam:EMPLOYEE/SALARY
Monitoring	L	newdata	ed.level	Integer	In	\$Root/XMLNSC/sam:EMPLOYEE/EDLEVEL
Monitoring						

12. Save the message flow <ctrl s>, keeping the message flow open in the Integration Toolkit.

3.3 Add Mapping Nodes

1. The new operation must make sure that the data that is sent to the R server is in the correct format. The rule was defined to operate on the EMPLOYEE schema, so the predictSalary operation must send the data in this format.

To do this, a Mapping node will be added to transform the incoming message from the predictSalary SOAP message to the EMPLOYEE schema format. A second Mapping node will be added to perform the reverse transformation.

In the predictSalary operation, add two mapping nodes, and connect them as shown.



2. Open the first mapping node, and click Next on the first dialogue window.

At the map inputs and outputs window:

- Select the input message = predictSalary
- Select the output message = EMPLOYEE

Click Finish.

🌐 New Message Map	
Select map inputs and outputs Creates a map that can contain message inputs and outputs with the Properties f LocalEnvironment can be added to the map after creation.	older. Optionally, database operations, message headers, and
Filter map input names (? = any character, * = any String):	Filter map output names (? = any character, * = any String):
Select map inputs	Select map outputs
	Comparison of the second state of the sec
Physical Location:	
Path: >> sample/libadmin/SAMPLE_EMPLOYEE.xsd Namespace: 1 http://sample/libadmin}	
?	< Back Next > Finish Cancel

3. Expand predictSalary, and map the input EMPLOYEE to the output EMPLOYEE.

Change the transform type to "Local map".

PredictSalary_Request_Response_Mapping						
▼predictSalary_Request	t_Response_Mapping	3 목 않 로 남 표 옥 두 ¥ 섹 섹 속 속 작 작				
E Message Assembly	predictSalary	□ 🕞 Message Assembly	EMPLOYEE			
⇒ <click filter="" to=""></click>		Section 2				
		Overrides	PropertiesType			
🗈 📌 Properties	[01] PropertiesType		EMPLOYEEType			
🖃 📌 predictSalary	[11] <anonymous></anonymous>					
EMPLOYEE	[11] EMPLOYEEType	local map 🗸 💆				

4. Click "Local map" to define the individual mappings.

Use the automap feature to map each input element to the same output element.

Save and close the map.

🔓 predictSalary_Request_Response_Mapping 🔹 🔊 EMPLOYEE							
🔻 🕶 👘 🔚 🐨 👘 📰 👘 📰 👘							
					5	£	
EMPLOYEE	EMPLOYEEType]			EMPLOYEE	EMPLOYEEType	
⇒ <click filter="" to=""></click>					⇒ <click filter="" to=""></click>		
e EMPNO	[11] <string></string>	🗰 Move 🔻			E EMPNO	[11] <string></string>	
E FIRSTNME	[1 1] <string></string>	Move 🗸			e FIRSTNME	[11] <string></string>	
	[ini] soungs	E Mayo -		\int	e MIDINIT	[01] <string></string>	
e MIDINIT	[01] <string></string>				LASTNAME	[11] <string></string>	
C LASTNAME	[11] <string></string>	+ Move -			e WORKDEPT	[01] <string></string>	
	[0 1] <string></string>	Move 🗸			e PHONENO	[01] <string></string>	
U WORKDEPT	[01] <suing></suing>				HIREDATE	[01] date	
C PHONENO	[01] <string></string>	→ Move →			e job	[01] <string></string>	
e HIREDATE	[01] date	+ Move -			e EDLEVEL	[11] short	

5. Open the second map, and click Next on the first dialogue window.

At the map inputs and outputs window:

- Select the input message = EMPLOYEE
- Select the output message = predictSalaryResponse

Click Finish.

Select map inputs and outputs Creates a map that can contain message inputs and outputs with the Properties LocalEnvironment can be added to the map after creation.	folder. Optionally, database operations, message headers, and
Filter map input names (? = any character, * = any String):	Filter map output names (? = any character, * = any String):
Select map inputs Select map Select map inputs Select map inputs Select map Select map Select map inpu	Select map outputs Select
Library: EmployeeServiceInterface Path: EmployeeService_InlineSchema1.xsd Namespace: Attp://EmployeeService}	< Back Next > Finish Cancel

6. Expand predictSalary, and map the input EMPLOYEE to the output EMPLOYEE.

Change the transform type to "Local map".

predictSalary_Request_Re	esponse_Mapping1		
predictSalary_Request	_Response_Mapping1	▙ \$\$\$ \$\$\$ \$\$\$ \$\$ \$\$ \$\$ \$\$ \$\$ \$\$ \$\$ \$\$ \$\$	0-
🖃 🛗 Message Assembly	EMPLOYEE	曰 টુ Message Assembly	predictSalaryResponse
$\Rightarrow i \rightarrow i $		$\rightarrow i \rightarrow i$	
		Overrides 🛛 😨 Mayo =	[01] PropertiesType
🗈 📌 Properties	[01] PropertiesType	E Arsian - C	se [11] <anonymous></anonymous>
			[11] EMPLOYEEType
EMPLOYEE	[11] EMPLOYEEType		
	[11] EMPLOYEEType	Local map - 2	

7. Click "Local map" to define the individual mappings.

Use the automap feature to map each input element to the same output element.

Save and close the map.

predictSalary_Request_	Response_Mapping 1 🎽 🚽	EMPLOYEE					
▼predictSalary_Reque	st_Response_Mapping1	a 6 6 4 4	· ↓ ► × ₽ 2		🏭 🛅 錔 💭 🛛 🕄		
						全	
EMPLOYEE	EMPLOYEEType]			🖃 🔬 EMPLOYEE	EMPLOYEEType	7
⇒ <click filter="" to=""></click>					⇒ <click filter="" to=""></click>		
C EMPNO	[11] <string></string>	Move 🔻]		e empno	[11] <string></string>	
	[1 1] catring >	Move -			C FIRSTNME	[11] <string></string>	
C PIRSTNIC	[1.1] <sung></sung>		-	\int	e MIDINIT	[01] <string></string>	
e MIDINIT	[01] <string></string>				C LASTNAME	[11] <string></string>	
e LASTNAME	[11] <string></string>	Move 🗸			C WORKDEPT	[01] <string></string>	
	[0 1] <string></string>	Move -			C PHONENO	[01] <string></string>	
WORKDEFT	[01] Sung>				C HIREDATE	[01] date	
C PHONENO	[01] <string></string>	+ Move +			e JOB	[01] <string></string>	
e HIREDATE	[01] date	+ Move -			e EDLEVEL	[11] short	

8. Save the updated message flow

4. Testing

4.1 Deploy the service

1. Save the updated message flow and deploy **both** the EmployeeServiceInterface library and the EmployeeService service in the usual way.



4.2 Test the service with the IIB flow exerciser

1. Open the **predictSalary Request_Response** subflow that you created earlier. Click the red button to start the flow exerciser,



2. The service will be deployed.

🌐 Progr	ress Information	×
1	Deploying [EmployeeService] to integration server [server 1] on integration node [IB 10NODE]	_
Details	>>	
	Close	

3. When the service has been deployed, a message box will appear as below, to show that the integration server is ready to record messages. Click on the **Close** button to continue.



Page 28 of 38 Implementing Analytics with the R Node using an Integration Service Version 10.0.0.0

4. Click the **send message** icon

🗺 EmployeeService 🛛		
S EmployeeService >	👹 predictSalary 🕨 🗉 Request_Response	× 🖓 1/1
👌 😳 Palette		
🔓 🖌 💷		
🙀 Favorites		
윾 WebSphere MQ	Input	
🖓 MQTT		
Gins JMS		
💭 НТТР	Mapping Mapping1	
🙀 Web Services	R	
0. SCA		

5. Click on **new message**

Send Message Send Message

Create or select a message to send to the flow. Click the message category header (e.g. Input Messages) for more information.

6. Select **predictSalary** under **Soap operation**, enter an employee number of '000010' and an education level of '19'. Then click **Send**. This will send a SOAP message to the predictSalary operation.



7. After a few seconds, an HTTP reply message should be received. Highlight and click on the notification.

۲	Progress Information	
	⊡ 🗄 Invoke Message Flow(new message 1)	
	Message flows deployment successfully completed	
	🖻 🖓 💦 Starting	
	Sending Message to "SOAP Input"	
	Received HTTP reply message for "SOAP Input"	
	Stopped	
		T
		Close

8. The HTTP reply details should be displayed in the lower pane. Scroll down to see the details.

Progress Information	
Invoke Message Flow(new message 1) Message flows deployment successfully completed Starting Sending Message to "SOAP Input" Received HTTP reply message for "SOAP Input" Stopped	
<pre></pre> xml version="1.0" encoding="UTF-8"? <soapenv:envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"> <soapenv:body> <out:predictsalaryresponse <br="" xmlns:io="http://sample/iibadmin">xmlns:out="http://EmployeeService"> <io:employee></io:employee></out:predictsalaryresponse></soapenv:body></soapenv:envelope 	Close

9. You should see that the record for employee number '000010' has been returned. However, the salary is the one predicted for the education level you supplied. Close the window when you are ready.

Progress Information	
Invoke Message Flow(new message 1) Message flows deployment successfully completed Starting	
 Sending Message to "SOAP Input" Received HTTP reply message for "SOAP Input" 	
Stopped	
<sex>SEX</sex> <birthdate>2002-01-01</birthdate> <salary>6.0439E+4</salary> <bonus>0</bonus> <comm>0</comm>	
	•
	Close

10. You will be returned to the EmployeeService view. Return the flow to edit mode by clicking on the red icon.

EmployeeService 🛛		
S EmployeeService >		🛛 🛱 🚺
		-
S EmployeeService	(I) EmployeeService	
SOAP/HTTP Binding	V getEmployee	
	W updEmployee	
L	Sector Salary	
	Error Handlers	
	Ba Failure	
	Ba Catch	
	B _□ <u>Timeout</u>	
		-
Service 🕕 Interface		

11. A warning message will pop up. Click on **Yes** to contine.



4.3 Understanding the value returned from R

The R node support in IIB V10 provides an interface for IIB to have the facility to call out to an R environment. Three scripting based interfaces are provided with the R node:

- 1) Connection script: Run when IIB initially connects to RServe.
- 2) Evaluate script: Run (per message) when a message reaches the R node.
- 3) Disconnect script: Run when the Integration Server stops or the application is redeployed.

We have used a very simple R model that predicts a value for SALARY based on a value for EDLEVEL. The data contained in the R model you used above, was based on an extract from the EMPLOYEE table in the DB2 SAMPLE database.

You will now use some provided R scripts to graphically represent how the model works and how the SALARY number was derived by the R model.

4.4 Analyze the model before and after running the application

You will now use R to show a graphical representation of the model before and after you run the application.

1. Open the R Gui interface (Start > All Programs > R > R x64 3.1.0).

RGui(64-bit) will open with an R Console.

File Edit View Misc Packages Windows Help	
R Console	×
<pre>R version 3.1.0 (2014-04-10) "Spring Dance" Copyright (C) 2014 The R Foundation for Statistical Computing Platform: x86_64-w64-mingw32/x64 (64-bit) R is free software and comes with ABSOLUTELY NO WARRANTY. You are welcome to redistribute it under certain conditions. Type 'license()' or 'licence()' for distribution details. Natural language support but running in an English locale R is a collaborative project with many contributors. Type 'contributors()' for more information and 'citation()' on how to cite R or R packages in publications. Type 'demo()' for some demos, 'help()' for on-line help, or 'help.start()' for an HTML browser interface to help. Type 'q()' to quit R. ></pre>	
<u> </u>	//.

2. In the console type "setwd("c:/student10/Analytics/RWork")"

(Note the Forward slashes).

The command sets the working directory for the R console session, you are setting the value of this working directory to where it can find scripts to run in the student10 directory.

If the command works there will be no response:

Ĩ	ABOUNDING MECHINE W. CONTROLS CONTROLS
ķ	<pre>setwd("c:/student10/Analytics/RWork")</pre>
þ	
\mathbf{V}	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

- - - -- ---

3. Type source "source ("DrawGraphs.r")" and press enter

4. A window with two graphs will open in RGui :



5. The first graph shows the original data that was built with the model.

The second graph shows a (Red) line. This is a (very simple) line of best fit to the data.

The (blue) mark at the red line shows how the value for SALARY was predicted given the EDLEVEL value of 19 (the SALARY value is taken from the Y (axis).

6. Go back to the flow exerciser and send a message with EDLEVEL=15.

Click the green arrow to execute the predictSalary. A value of 4.524E+4 will be returned in the SALARY field:

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~		
	HIREDATE	2001-01-01
	JOB	}
	EDLEVEL	15
1	SEX	F
	BIRTHDATE	2001-01-01
	SALARY	4.524E+4
	BONUS	
hana	COMM	A

7. Switch back to the R console and re run the DrawGraphs script

The command is **source ("DrawGraphs.r")** (highlight the R console and press the up arrow to retrieve previous commands) 8. A blue mark will now appear on the line (above 15 on the X axis).



Again the value passed back to the message flow was taken from the Y axis of the graph.

## END OF LAB GUIDE