

IBM Integration Bus

Creating an Integration Service

Featuring:

Creating schemas from a database Placing the schemas in a Shared Library Using the schemas in an Integration Service Placing a Mapping Node in a Shared Library Referencing a Submap in a Shared Library Using the Submap to read a database Introduction to the Flow Exerciser

January 2016 Hands-on lab built at product Version 10.0.0.3

1. INTRODUCTION	3
 1.1 SCENARIO OVERVIEW 1.2 OUTLINE OF TASKS 1.3 CONFIGURE INTEGRATION BUS NODE TO WORK WITH DB2	4 5 6
2. CREATE DATABASE DEFINITION, SCHEMAS AND SUBMAP	7
 2.1 CREATE A DATABASE DEFINITION FOR HRDB	13 17 27
3. CREATE THE INTEGRATION SERVICE	40
 3.1 IMPLEMENT THE MAIN MAP 3.2 IMPLEMENT THE GETEMPLOYEE OPERATION 	47 53
4. TEST THE INTEGRATION SERVICE WITH THE FLOW EXERCISER	55
5. TEST THE SERVICE WITH SOAPUI	66
END OF LAB GUIDE	69

1. Introduction

An integration service is a specialized application with a defined interface and structure that acts as a container for a Web Services solution.

You can integrate applications by using a service-oriented architecture (SOA). In an SOA, a service is often defined as a logical representation of a repeatable activity that has a specified outcome. Typically, a service is self-contained and its implementation is hidden from its consumers. To facilitate their reuse, services define a prescribed interface, which specifies how data is exchanged with the service.

In IBM Integration Bus, an integration service is a specialized application with a defined interface that acts as a container for a Web Services solution:

- It contains message flows to implement the specified web service operations.
- The interface is defined through a WSDL file.

Important note

This lab, version 10.0.0.3, has been updated significantly from earlier versions. The following changes have been made:

You should use the Windows user "iibuser". This user is a member of mqbrkrs and mqm, but is not a member of Administrators. The user "iibuser" can create new IIB nodes and do all required IIB development work. However, installation of the IIB product requires Administrator privileges (not required in this lab).

The database has been changed from the DB2 SAMPLE database to the DB2 HRDB database. HRDB contains two tables, EMPLOYEE and DEPARTMENT. These tables have been populated with data required for this lab. (The DDL for the HRDB is available in the student10 folder; we intend to provide corresponding DDL for Microsoft SQL/Server and Oracle over time).

The map node now retrieves multiple rows from the database, using an SQL "LIKE" function . Additionally, the map has been refactored to use a main map and a submap. Both the main map and submap are located in a shared library.

Input to the integration service is now a simple schema containing just one element, the required employee number.

As a consequence, this version of the lab, and the associated solution, can only be used with the corresponding changes in other labs. Use version 10.0.0.3 of all labs in this series of lab guides.

1.1 Scenario Overview

In this lab, you will create an integration service and all the IBM integration resources required to implement the service. You will define the service data model and the WSDL that describes the service. You will implement and test the service operation.

The service accesses an external database and responds with data retrieved from the database. You will need to configure IBM Integration Bus to work with an external database system. In this lab, we are using DB2.

1.2 Outline of tasks

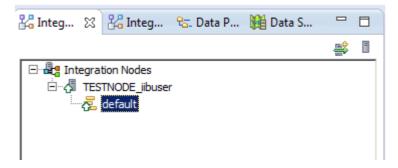
The tasks to complete in this lab are the following:

- 1. Configure the IBM Integration Bus default runtime node to work with DB2
 - a. Runtime configuration: Configure a JDBCProvider configurable service. Configure the security credentials to work with the database.
 - b. Development configuration: Create a Database Definition project, and the database connectivity configuration.
- 2. Create a Shared Library. This library will contain your message models, the WSDL that defines the new service, and the map and submap for the service.
- 3. Create a Message Model
 - a. Create a message model based on a database table definition
 - b. Extend the message model with other schema available in the library
- 4. Create an integration service, and define the WSDL associated with it
- 5. Implement the integration service operation
 - a. Create the Submap that retrieves employee data from the EMPLOYEE table
 - b. Create the main Message Map in the Shared Library
- 6. Test the service using the Integration Toolkit Flow Exerciser
- 7. Test the integration service using external tools (optional)
 - a. Test the service using SOAPUI

1.3 Configure Integration Bus node to work with DB2

- 1. Login to Windows with the user **iibuser**, password = **passw0rd**.
- 2. Start the IIB Toolkit, using the shortcut on the Windows Start menu. Accept the default name for the workspace (or you can specify your own location if desired).

When you start the Toolkit, a new IIB node named TESTNODE_ibuser will be created. This node will be started automatically when the Toolkit starts. You will be able to see this node in the Integration Nodes view of the Toolkit.



If the IIB Healthcare Pack product is installed, you will also see the Configurable Services folder.

3. To run this lab, the Integration Bus node must be enabled to allow a JDBC connection to the HRDB database.

Open an IIB Command Console (from the Start menu), and navigate to

c:\student10\Create_HR_database

4. Run the command

3_Create_JDBC_for_HRDB

Accept the defaults presented in the script. This will create the required JDBC configurable service for the HRDB database.

5. Run the command

4_Create_HRDB_SecurityID

6. Stop and restart the node to enable the above definitions to be activated

mqsistop TESTNODE_iibuser

mqsistart TESTNODE_iibuser

1.3.1 Recreating the HRDB database and tables

This section will not normally be required if you are running the lab scenarios on the pre-built VMWare workshop image. If you are creating your own environment, you will need to run the commands in this section. You may need to alter the database schemas and authorities in the DDL to reflect your own environment.

The HRDB database, and the EMPLOYEE and DEPARTMENT tables have already been created on the supplied VMWare image. If you wish to recreate your own instance of this database, do the following tasks:

- 1. Login with the user iibadmin (password=passw0rd)
- 2. Open an IIB Command Console (from the Start menu), and navigate to

c:\student10\Create_HR_database

3. Run the commands

1_Create_HRDB_database

2_Create_HRDB_Tables

4. Logout user iibadmin.

Appropriate database permissions are included in the scripts to GRANT access to the user ibuser.

1.4 Open the Windows Log Monitor for IIB

A useful tool for IIB development on Windows is the IIB Log Monitor. This tool continuously monitors the Windows Event Log, and all messages from the log are displayed immediately. This tool is no art of the IIB product, and is currently only available with the IIB workshop system.

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
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]
BIP21551: (IBIØNODE.server1) About to 'create '' the deployed resource ''gen.
getEmployee_EmployeeService_EmpServClient_JSON1'' of type '.SUBFLOW''. [10/3/20
14 5:07:37 PMJ
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
22014 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

2. Create Database Definition, schemas and Submap

The IIB environment is now prepared, so move on to creating the artefacts that will be used in this lab. The first artefact is a Database Definition that represents the HRDB database.

2.1 Create a Database Definition for HRDB

The first artefact is a Database Definition that represents the HRDB database. The database definition is required to enable the IIB Toolkit to be able to connect to the database, so that it can extract the required table and schema definitions, and use these to create an IIB message model.

Make sure you are logged in as "<u>iibuser</u>", and open the IIB Toolkit using the shortcut on the Windows Start menu.

🖧 Patterns Explorer 🖶 Application Dev... 🔀 ∇ Application Development New... New Application... New Integration Service... New REST API ... New Library... New Ressage Flow B Subflow Message Model ... 🚖 Message Map Move.... ESQL File Rename... 🔛 Broker Schema 🔄 Import... Adapter Connection Database Definition 👍 Export... н 🖹 Data Lineage Documents Refresh BAR file Decision Service MQ Service

🕼 Database Service

1. In the Toolkit navigator, right-click in blank space, select New, Database Definition.

2. Click New to create a data design project.

🌐 New Database D	efinition File				
Create a database Specify the database	e definition file e type, version and the data design projec	t that will contain the da	atabase definition fi	ile.	
Data design project: Database:	DB2 for Linux, UNIX, and Windows				New
Version:	V10.1				
?		< Back	Next >	Finish	Cancel

3. Name it HRDB. Click Finish.

lev Data Design Project	그미스
Create a data design project	
Specify a basic definition for the new project. This project stores data design objects.	T
Project name: HRDB	
✓ Use default location	
Location: C:\Users\ibuser\IBM\IIBT10\workspace2\HRDB Brow	se,,,
└─ Working sets	
Add project to working sets	
Working sets: Select	
Finish Ca	ancel

4. Click Next.

🌐 New Database D	efinition File	
Create a database Specify the database definition file.	e definition file e type, version and the data design project that will contain the database	
Data design project: Database: Version:	HRDB DB2 for Linux, UNIX, and Windows 💌 V10.1 💌	▼ <u>New</u>
?	< Back Next > Finish	Cancel

5. Click New to create a new connection.

🜐 New Database Definitior	File				_ 🗆 🗙
Select Connection Select an existing connection.					
Connections					
- Descetion					New Edit Delete
Properties Property	Value				
?		< Back	Next >	Finish	Cancel

6. Set the Database name to HRDB.

Specify connection credentials (user=iibuser, password=passw0rd),

If you wish, you can click Test Connection, to make sure you have specified the connection details correctly.

Optionally, select save the password, and click Finish. (this is only required for subsequent Toolkit access to the database connection; it is not used by the IIB runtime).

(You may need to expand the dialog window to see the Save Password checkbox).

New Connection	
onnection Parameters Select the database manager and a JDBC driver, and specify required connection parameters.	
Connection identification Image: Connection Name: Image: Ima	
< Back Next > Einish Cancel	

7. Highlight the new connection (HRDB1), and click Next.

New Database Definit					
Select an existing connect	ion.				FB
Connections					
HRDB1					New
					Edit
					Delete
 Properties 					
Property	Value				
Name	HRDB1				
Description					
Category	Database Connect	ions			
Database	HRDB				
JDBC Driver Class	com.ibm.db2.jcc.D	B2Driver			
Class Location	C:\IBM\IIB\10.0.1	075.3\tools\plug	ins\com.ibm		
Connection URL	jdbc:db2://localho	st:50000/HRDB:	retrieveMes		
User ID	iibuser				
,	'			-	
?		< Back	Next >	Finish	Cancel

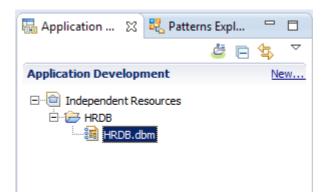
8. Select the iibadmin schema, and click Finish.

(Note that the HRDB database has been defined with the IIBADMIN schema. "IIBUSER" has been granted permissions on this database. For reference, this is shown in the script 2_Create_HRDB_Tables.cmd).

🌐 New Database Definition File				
Select Objects Select schema(s) to reverse engineer.				
Apply name filter (? = Any character, * = Any string	g):			Clear Filter
Select objects:				
☑ IIBADMIN ☑ IIBUSER ☑ NULLID ☑ SQLJ ☑ SYSCAT ☑ SYSFUN ☑ SYSIBM ☑ SYSIBMADM ☑ SYSIBMINTERNAL ☑ SYSIBMIS ☑ SYSPROC ☑ SYSTAT				Select All
(?)	< Back	Next >	Finish	Cancel

9. The database definition for the HRDB database is now present as a project in the Independent Resources part of the navigator.

Note that this HRDB.DBM file can be used by any other developer requiring access to the database. It contains a definition of the database, and the selected tables and schemas, and can be shared as required, to create the related IIB schema definitions.



2.2 Create the IIB Message Model (XML schema)

The message model schema will be derived from the HRDB database definition that you have just created. This lab will use the EMPLOYEE table, but you will generate schemas for both the EMPLOYEE and DEPARTMENT tables, for future use in later labs.

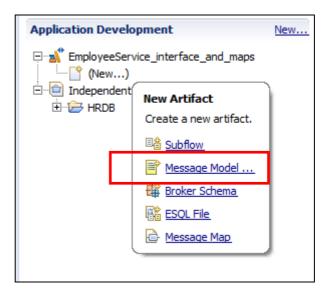
1. A Message Model is defined within a container called a Library, so you must first create a Library. In the Navigator view, click New, and select "Start by creating a library".

Nev	N	
	New Artifact	
	Message Flow	
	E Subflow	
	Message Model	
	Amessage Map	
	ESQL File	
	Decision Service	
	MQ Service	
	Database Service	
	Quick Start	
	Start by creating an application	
	Start by creating an integration service	
	Start by creating a REST API	
	🛋 Start by creating a library	
	Start from WSDL and/or XSD files	
	Start by discovering a service	

2. You can choose whether to create a Shared or Static library. Choose Shared (the default), and name the library **EmployeeService_interface_and_maps**.

Click Finish.			
🌐 New Library			
Create a new library Create a library to group reusable resource together. Enter a name for the new library		an be shared and manag	ged 🛋
Library name EmployeeService_interface	e_and_maps		
Select Shared Library when you require or more applications. When a shared lib applications or shared libraries that refer	rary is updated,		
?	< Back	Next > Fir	nish Cancel

3. In this library create a new Message Model (click "New" under the library name), then select "Message Model".



4. Select the Database Record type, and click Next.

🕟 New Message Model	
Create a new messag	je model file 📃 👘
Select the message model ty	be or format
XML	
C SOAP XML	XML data for use in Web Services.
O Other XML	All other XML data.
Text and binary	
C CSV text	Comma Separated Values data, a delimited text format commonly used as an export format by spreadsheets and databases.
C Record-oriented text	Text data formats where delimited fields are grouped into records.
C COBOL	Data for COBOL programs
О с	Data for C programs
\bigcirc Other text or binary	All other text or binary data formats.
Enterprise Information Sy	
O SAP	Data from SAP systems including IDoc and BAPI
O Siebel	Data from Siebel systems
O PeopleSoft	Data from PeopleSoft
O JD Edwards	Data from JD Edwards systems
Other	
C CORBA IDL	Data from CORBA
Oatabase record	Records from relational databases
	Data for extended email format
C IBM supplied	Predefined data format
(?)	< Back Next > Finish Cancel
0	

5. Create an XML schema - click Next.

😰 New Message Model				
Database record Choose how you would like to create your message model fro	m a database de	efinition file.		
Integration Bus can access user databases to manipulate the	business data.			
• Create an XML schema file from a database definition.				
?	< Back	Next >	Finish	Cancel

6. Select the HRDB.dbm file from the database definition project in the workspace (under Independent Resources).

This will automatically create the XML Schema file name, HRDB.xsd.

Click Next.

🌐 New Message Mod	el				
	ge model file from a databas library project to contain the new me		a database .dbm	file to import.	
Application or Library:	EmployeeService_interface_and_mag	ps		•	New
Folder	<specifying a="" folder="" is="" optional=""></specifying>				Browse
XML Schema file :	HRDB.xsd				
 Select file from work 	space:				
Select file from outsi	B IRDB.dbm ide workspace: efined database types as XML schema	12		Y	Browse
?		< Back	Next >	Finish	Cancel

7. Select the HRDB database, which will automatically select the EMPLOYEE and DEPARTMENT tables.

Click Next, then Finish.

🌐 New Message Model				_ 🗆 🗙
Create a new message model file from a database Select the tables from which you want to generate message				
Select the tables to import:				
?	< Back	Next >	Finish	Cancel

8. The schema will be created, and the Toolkit will open it with the XML schema editor. Note that the name of the schema is HRDB.xsd.

		View: Advanced
S Schema : H	http://hrdb/iibadmin	
Elements	Types	
Elements	EPARTMENTType	
DEPARTMENT : DEPARTMENTType		
DEPARTMENT : DEPARTMENTType		
DEPARTMENT : DEPARTMENTType		

2.3 Add additional references to the schema

This lab will use a database scenario to illustrate the use of the Graphical Mapping Node within an Integration Service.

To do this, the Integration Service will pass a return code, plus details of any SQL messages, to the requesting application. You will therefore import a second schema, DBResp, into the HRDB schema.

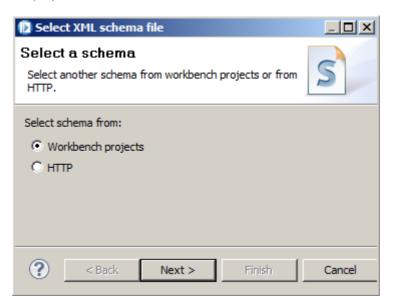
The DBResp schema contains several elements that can be used to report SQL responses, as well as a user return code.

The HRDB schema will be augmented with a new complex type, EmployeeResponseType, which contains the EMPLOYEE and DBResp types.

1. In the HRDB schema, right-click in the Directives section.

Click Add Import. Schema : http://hrdb/iibadmin 🛵 Directives Add Include Add Import Add Redefine 尾 Elements 🕮 Types Show properties DEPARTMENT : DEPARTMENTType ENTType Refactor • EMPLOYEE : EMPLOYEEType EType References Þ Π

2. Select Workbench projects, and click Next.



3. Click Import Files.

lect XML schema file	<u>_ 0 ×</u>
Select XML schema file Select an XML schema file from the Workbench projects	S
Workspace Files	E E
Import Files	
Reck Next > Finish	Cancel

4. In the "From directory", use the browse button to navigate to

```
c:\student10\integration_service\resources
```

Select the **DBResp.xsd** schema.

Use the second Browse button to select the target folder:

```
EmployeeService_interface_and_maps
```

Click Finish, and then Finish again on the next window.

🌐 Import	_O×
File system Import resources from the local file system.	
From directory: C:\student10\Integration_service\resources	Browse
····· ♥ ➢ resources ♥ INDBResp.xsd ■ INDB.zip HRDB.zip	
Filter Types Select All Deselect All	
Into folder: EmployeeService_interface_and_maps	Browse
Options	
Create top-level folder Advanced >>	
? Finish	Cancel

5. Note that the DBResp schema has been imported.

In the Types pane, right-click in an open area, and select Add Complex Type.

S Schema : http://hrdb/iibadmin						
🛵 Directives						
4//DBResp.xsd						
Elements	Types					
DEPARTMENT : DEPARTMENTType	DEPARTMENTType					
EMPLOYEE : EMPLOYEEType	EMPLOYEEType					
	Add Complex Type					
	Show properties					
	Refactor					
	References					

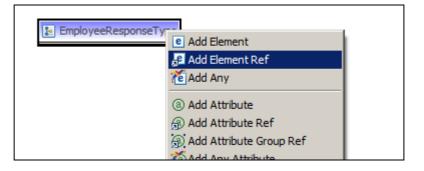
6. Set the name of the new Complex Type to EmployeeResponseType.

Schema : http://hrdb/iibadmin				
Directives				
4//DBResp.xsd				
Elements	Types			
DEPARTMENT : DEPARTMENTType EMPLOYEE : EMPLOYEEType	DEPARTMENTType Image: EmployeeResponseType Image: EMPLOYEEType			

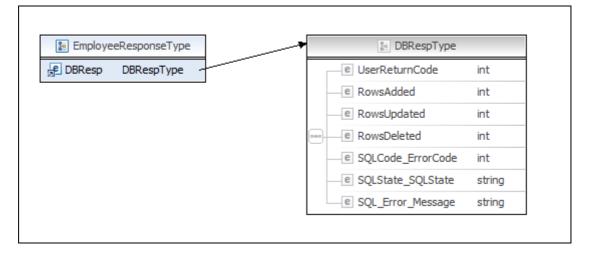
7. Double-click EmployeeResponseType which will open the schema editor for this type.

🕄 *HRDB.xsd 🔀
EmployeeResponseType

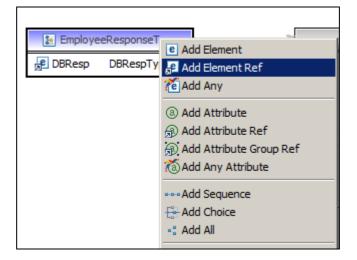
8. Right-click EmployeeResponseType and select Add Element Ref.



9. The editor will add a default element using the first alphabetically available. Luckily, since DBResp is the first available, this is the one that we want.



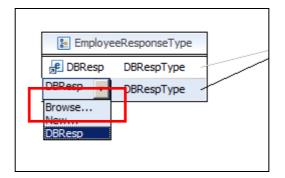
10. Now, add a second element reference, by right-clicking the EmployeeResponseType again.



This will add a second reference to DBResp.

🔚 Employe			
📌 DBResp	DBRespType		\sim
📌 DBResp	DBRespType	~	

11. To change this second reference, click the second instance of DBResp (click DBResp, not DBRespType). A drop-down menu will be shown. Click Browse.



12. Highlight the EMPLOYEE element, and click OK.

Set element reference	×
Name (? = any character, * = any string):	
Elements:	
C DBResp	
C DEPARTMENT	
EMPLOYEE - http://hrdb/iibadmin	
Declaration Location:	
S /EmployeeService_submaps/hrdb/iibadmin/HRDB.xsd	
⊂ Search Scope	1
O Workspace O Enclosing Project Current Resource	
O Working Sets Choose	
OK Cancel	1
	1

13. Since multiple rows might be returned from the EMPLOYEE table, it will be necessary to be able to handle multiple instances of the EMPLOYEE element within EmployeeResponse, so set the Multiplicity as follows:

Right-click EMPLOYEE, click "Set Multiplicity", then "0...* (Zero or more)".

🔚 Employe	eResp	oonseType				B DBRespType
🚽 DBResp		DBRespType			e Us	serReturnCode
 EMPLOYEE [[11]	Insert Element	•	e RowsRetrieved		owsRetrieved
		Set Type	\rightarrow		e Ro	owsAdded
		Set Multiplicity	•	11 (Requi	ired)	wsUpdated
		💢 Delete	_	01 (Optio		wsDeleted
				0* (Zero (LCode_ErrorCode
		Show properties		1* (One o		LState SQLState
		Refactor	•			QL_Error_Message
		References				

When you've done this, the complex element will look like this (click the EMPLOYEE element) (You may need to click in an open area of the editor to update the display of the multiplicity).

EmployeeResponseType				
	DBResp		DBRespType	-
	EMPLOYEE	[0*]	EMPLOYEEType	

The properties will look like this (note the minimum and maximum occurrences when the EMPLOYEE element is selected):

🔲 Properties 🔀 💦 Problems 🚦 Outline 🧔 Tasks 🥅 Deployment Log						
🔊 element ref						
General	<u>N</u> ame:	EMPLOYEE				
Documentation	Reference:	hrdbiibadmin:EMPLOYEE				
Extensions	<u>Type</u> :	hrdbiibadmin:EMPLOYEEType				
Advanced <u>Minimum Occurrence</u> :		0				
	Maximum Occurrence:	unbounded				

14. Return to the high-level definition of the schema (click the highlighted icon shown below).

The Mapping Node bases its message inputs and outputs on element definitions, rather than element types, so it is necessary to define a new element named EmployeeResponse.

In the Elements pane, right-click, and select "Add Element".

S *HRDB.x	sd 🛛	
		View: Advanced 🗢
	Schema : http://hrdb/	iibadmin
	//DBResp.xsd	
	E EMPLOYEE : EMPLOYEET ype	PARTMENTType poloyeeResponseType IPLOYEEType

15. Name the new element EmployeeResponse.

S Schema : http://hrdb/iibadmin				
🚝 Directives				
4//DBResp.xsd				
Elements	Types			
DEPARTMENT : DEPARTMENTType				
E EMPLOYEE : EMPLOYEEType	EmployeeResponseType			
e EmployeeResponse : string	EMPLOYEEType			

16. Set the Type of the new element to EmployeeResponseType (right-click, Set Type, Browse).

🕒 Elements				
EMPLOYEE : EMPLOYEEType				
EmployeeRespons	Set Type Set Multiplicity		New Browse	
			browse	

Type "e" into the filter to reduce the list. Select EmployeeResponseType, and click OK.

🌐 Set Type	×
Name (? = any character, * = any string):	
e	[
Types:	
EmployeeResponseType - http://hrdb/iibadmin E EMPLOYEEType E ENTITIES E ENTITY	
Declaration Location:	
/EmployeeService_interface_and_maps/hrdb/iibadmin/HRDB.xsd	ł
Search Scope	
C Workspace C Enclosing Project C Current Resource	
C Working Sets Choose	
OK Cancel	

Save and close the HRDB.xsd schema file.

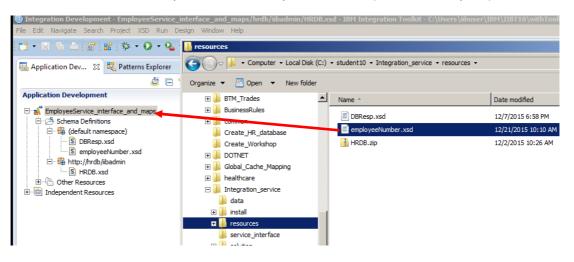
2.4 Import employeeNumber Schema

Finally, we need an element (and element type) that represents the employee number as a single element.

1. For the purposes of this lab, a suitable schema has been provided, so copy/paste (or drag/drop) this into the shared library.

```
Use the file
c:\student10\integration_service\resources\employeeNumber.xsd
```

This schema has no namespace, so will be imported into the (default namespace).



2.5 Create the submap

The new service will contain just one operation. This operation will use a Mapping Node to retrieve employee details from the EMPLOYEE table in the HRDB database. The function to access the database will be isolated in a submap, so that it can be invoked from a variety of applications. The submap will be created and stored in the Shared Library.

1. Highlight the EmployeeService_interface_and_maps library, and click New. Select Message Map.

🔚 Application Develo 🔀 💐 Patterns Explorer 🗧	⊐ 🗖 🛐 *HRDB.xsd 🔀
🖉 📄 🔄	New
EmployeeService_interface_and_maps Schema Definitions Government (default namespace) JoBResp.xsd Government Structure http://hrdb/ibadmin Structure Structure http://www.example.org/employeeNumber SemployeeNumber.xsd Government Other Resources Independent Resources	New Artifact Image: Message Flow Image: Subflow Image: Message Model Image: Message Mape Image: Message Model Image: Message Message Model Image: Message Mess
⊟… 🔁 HRDB 🗃 HRDB.dbm	Database Service Quick Start
	 Start by creating an application Start by creating an integration service Start by creating a REST API Start by creating a library Start from WSDL and/or XSD files Start by discovering a service

- 2. Set Type of map = "Submap called by another map".
 - Make sure the Container is EmployeeService_interface_and_maps
 - Set the Map name to "getEmployees_submap"
 - Double-check that you have selected Type = "submap".

Click Next.

	🌐 New Message Map	_ 🗆 ×
	Specify a new message map file Select map type, container, name, and broker schema for the new map.	
\langle	Type of map that you want to create: C Hessage map called by a message flow node © Submap called by another map	
(Container: EmployeeService_interface_and_maps Map name: getEmployees_submap	New
	Map organization Image: Way organization Image: Way organization Image: Way organization	
	Schema: (default broker schema)	*
	Sack Next > Finish	Cancel

3. Expand the EmployeeService_interface_and_maps library for both input and output to the map. Expand DFDL and XML Schemas.

Since this is a submap, you should use only "Type" definitions, not elements.

- For the input, select the input message "employeeNumberType".
- For the output, select the output message "EmployeeResponseType".

Click Finish.

🌐 New Message Map	
Select map inputs and outputs Creates a map that can contain message inputs and outputs. Optionally, creation.	, database operations can be added to the map after
Filter map input names (? = any character, * = any String):	Filter map output names (? = any character, * = any String):
e	e
Select map inputs	Select map outputs
Library: 2 EmployeeService_interface_and_maps Path: 2 hrdb/iibadmin/HRDB.xsd Namespace: 1 {hrdp://hrdb/iibadmin}	
•	< Back Next > Finish Cancel

4. The map editor will be shown, but no transforms have been made so far.

Click the "Select rows from a database" icon.	
GetEmployees_submap ☆	
getEmployees_submap	
▼getEmployees_submap 🖉 🖗 🐉 🕼 🎶 🕼 🗶 🍣 🛱	⁰ ѷ ѷ 🔲 🖩 🖮 🏭 🕮 💭
िं employeeNumberType	EmployeeResponseType
	$\stackrel{\geq I_{i}}{\rightarrow I^{\circ}} < Click to filter>$
	■ DBResp [11] DBRespType
	■ P EMPLOYEE [0*] EMPLOYEEType

5. Since the shared library does not have a project reference to the HRDB.dbm project, click the "Add database" link.

Define a where clause		
a key column in the database t evaluate to a boolean. ise	able. The value can come from o	ther inputs in the map. The expression must
Table columns	Operators	Available inputs for column values
		SteessageAssembly Stepson Stees Stepson Stees Stepson Steep St
SQL where dause		<u> </u>
Plac XPath expression	1	Edit Add Remove
	SQL where dause SQL where dause 1=1	The where clause is used to extract only those rows that fulfill a key column in the database table. The value can come from or evaluate to a boolean. Table columns Operators AND OR

6. Select the HRDB database, and "Make available".

When HRDB appears in the pane on the lower right, click OK.

🌐 Add Database			×
Add a database definition model	file		
Choose a method to make a dbm file available to the n	nap. A dbm file defines a data	model.	
Import from a database			
Connect to a database and select a physical data me	odel.		
Import			
Import a data design project			
Import a data design project into the workspace. Fo getting a project from a repository.	r example, you may obtain a	data model by importing a project interchange or	
Import			
Make a data design project available from th			
Some data models may reside in projects that are cu available.	rrently unavailable from the r	nap. Select a project from the list below to make it	
Data design proj <mark>e</mark> cts unavailable to the map		Data design projects available to the map	
HRDB	Make available >		-
	Make all available >>		
1		1	
?		ОК	

- 7. In the Database Select wizard:
 - Select (tick) the EMPLOYEE table (note this will select all columns you can select individual columns if you want). Note - only select the EMPLOYEE table, not the DEPARTMENT table. This function is the generating the SQL SELECT clause.
 - 2. Remove the "1=1" phrase from the "SQL where clause" pane.
 - 3. Construct the SQL where clause:
 - a. Double-click the EMPNO column
 - b. Double-click the "LIKE" operator
 - c. Double-click the \$employeeNumberType element from the available inputs

As a check, the SQL clause that should be generated should be:

IIBADMIN.EMPLOYEE.EMPNO LIKE ?

And the generated XPath expression should be:

\$employeeNumberType

hoose a database to select from	Define a where clause			
elect a database available to the map, or obtain a ifferent database.	The where clause is used to extract on a key column in the database table. Th evaluate to a boolean.			
	Table columns	Operators	vailable inputs for column	values
hoose the columns to include ou must choose at least one column.	SQL where dause	▲ AND → OR → NOT → = → < → > → = → = → = → = → = → = ↓ → = ↓ → = ↓ → = ↓ → = ↓ → = ↓ → ↓ → ↓ → ↓ → ↓ → ↓ → ↓ → ↓ → ↓ → ↓ →	SemployeeNumbe	/Type
	IIBADMIN.EMPLOYEE.EMPNO LIKE		Edit	
Classify SQL warning				Add
	•			Remove

8. Refine the XPath expression.

The map will be built to retrieve all employee records that partially match the provided key. For example, if the provided key is "0020", we want the map to retrieve rows with a value of "000020", "00201", "00204", etc.

This is done by extending the SQL LIKE statement, in conjunction with the "%" character, appended both as a prefix and a suffix. Hence the statement "SELECT xxxx LIKE %0020%" will achieve the result described above.

To do this, edit the XPath expression as follows:

```
fn:concat('%', $employeeNumberType, '%')
```

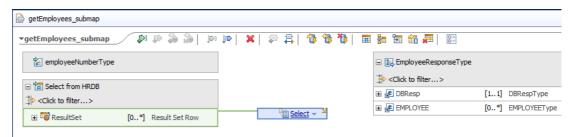
The result should look like this:

SQL where	dause		
IIBADMIN	EMPLOYEE.EMPNO LIKE ?		×
Plac	XPath expression fn:concat('%', \$employeeNumberType, '%')	Edit	Add Remove

Now double-check that you typed this XPath exactly as shown above !!

Click OK to close the database definition window.

9. The map will be shown as follows. You will now see the Select input assembly showing in the map editor.



(Note - you can return to refine the SQL statement by hovering over and selecting the database icon, as below).

Select from HRDB			
🗈 🐻 ResultSet	[0*]	Result Set Row	-

10. In the output message assembly, connect the Select transform to EmployeeResponseType.

Then, click "Select" to enter the next logical level of the map.

C employeeNumberType					_		mployeeResponseT	уре	
Select from HRDB						🔆 <clic< td=""><td>ck to filter></td><td></td><td></td></clic<>	ck to filter>		
☆ <click filter="" to=""></click>						🕀 📌 DE	BResp	[11]	DBRespType
			Ę	Select		🕀 📌 EN	MPLOYEE	[0*]	EMPLOYEETyp
⊞ 🐻 ResultSet [0*] Result Set	tRow								

11. Connect ResultSet to EMPLOYEE in EmployeeResponseType. This will generate a "For each" transform.

getEmployees_subma	ap 🔊 🖓 🖓 👸	व्या व्य	X 🗧 🛱	1 8 1) 🖩 🏚 🛱 🛱 🚚	0- 0-
					<u></u>	
🖃 📸 ResultSet	[0*] Result Set Row		For each 👻 🎽		🖃 🔀 EmployeeResponseType	
⇒ <click filter="" to=""></click>					⇒ <click filter="" to=""></click>	
EMPNO	[11] CHAR				🗄 📌 DBResp	[11] DBRespType
FIRSTNME	[11] VARCHAR				EMPLOYEE	[0*] EMPLOYEEType
MIDINIT	[11] CHAR					
LASTNAME	[11] VARCHAR					
WORKDEPT	[11] CHAR					
PHONENO	[11] CHAR					

12. The "For each" transform will indicate a Warning, and suggest a "quick fix". Hover over the light-bulb on the transform, which will show the available "quick fix".

The For each transformation does not	
1 guick fix available:	
Click a fix to invoke	

.

13. Invoke the quick fix by clicking "Invoke automap". Since we simply want to map similarlynamed elements, you can accept the defaults here, and click Finish.

🌐 Auto Map	
Automatically map inputs to outputs	=
Choose the options to automatically map the selected input and output elements. Click Next to select the transforms to create, or click Finish to create the transforms for all the matching elements.	
Mapping Scope	_
• Map all simple descendants of the selected elements	
Group transforms into nested maps	
C Map the immediate children of the selected elements	
Name Matching Options	
Case sensitive	
Alphanumeric characters (Letters and digits only)	
Mapping Criteria	
Press F1 for more information when the names of inputs and outputs satisfy more than one criterion.	
Create transforms when the names of inputs and outputs are the same	
Create transforms when the names of inputs and outputs are more similar than	
▼ 60 % match	Default
Create transforms when the input and output names are matched to synonyms defined in a file	
	Browse
	_
Back Next > Finish	Cancel

14. The Warning will have been resolved.

			企	
∃ [®] ResultSet	[0*] Result Set Row	For each ▼ ⁴		
EMPNO	[11] CHAR		🖃 📌 DBResp	[11] DBRespType
FIRSTNME	[11] VARCHAR		e UserReturnCode	[11] int
MIDINIT	[11] CHAR		e RowsRetrieved	[11] int
LASTNAME	[11] VARCHAR		e RowsAdded	[11] int
WORKDEPT	[11] CHAR		e RowsUpdated	[11] int
PHONENO	[11] CHAR		e RowsDeleted	[11] int
HIREDATE	[11] DATE		e SQLCode_ErrorCode	[11] int
JOB	[11] CHAR		e SQLState_SQLState	[11] string
EDLEVEL	[11] SMALLINT		e SQL_Error_Message	[11] string
SEX	[11] CHAR	C		[0*] EMPLOYEEType
BIRTHDATE	[11] DATE			
SALARY	[11] DECIMAL			

15. We want to set the value for RowsRetrieved.

Expand DBResp, and connect the input ResultSet to the output DBResp/RowsRetrieved. This will initially create a "For each" transform; the transform will show a warning.

						£	
🖃 🖓 ResultSet	[0*]	Result Set Row		E For each ▼ ²⁴	EmployeeResponse → <click filter="" to=""></click>		EmployeeRespor
> ≪Click to filter>					🖃 📌 DBResp	[11]	DBRespType
	[11]	CHAR			e UserReturnCode	[11]	int
FIRSTNME		VARCHAR			RowsRetrieved	[11]	int
MIDINIT	[11]				e RowsAdded	[11]	int
			-		e RowsUpdated	[11]	int
LASTNAME	[11]	VARCHAR			e PowsDeleted	[1 1]	int

16. Click the drop-down arrow on the transform, and change the transform to "Custom XPath".

Eor each V EmployeeResp	ons
For each - Honore Click to filter	>
▼	F
If	
Select	
Task	ve
	Г
Custom Fallstoffis	E.
	ed
Custom Java	d
S Custom XPath	ITTC
E String Functions	Ë
😟 🖄 Boolean Functions	QL
中	Me
Allows you to enter your own XPath 🔺	
expressions in the property pages to be used in the transform.	
to be used in the transform.	
	1

17. In the map editor, make sure the Custom XPath transform is selected.

On the Properties tab for this transform, select "General" and type the following into the XPath editor:

fn:count(

Then, invoke the content assist function (Ctrl-space) which will show you the possible values for completion. Select the value that shows \$ResultSet, or similar. Depending on whether you have made other editing changes, the value shown may be \$ResultSet or have a suffixed number. The example shown below is \$ResultSet.

Complete the XPath expression with a ")". This expression will calculate the number of rows retrieved from the database.

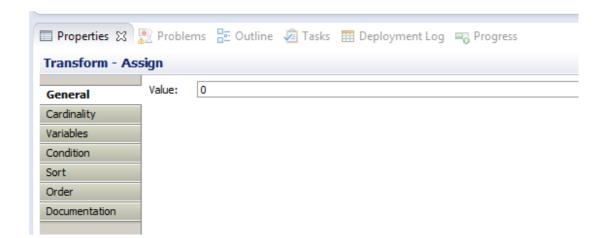
Properties 🔀 🚦	Problems	🗄 Outline 🖉 Tasks 🔠 Deployment Log
Transform - Cus	tom XPath	
General	fn:count(Insert Simple XPath
Namespaces	m:count	·····XY fn:abs (numeric?) : numeric?
Cardinality		xy fn:adjust-dateTime-to-timezone (xs:dateTime?) : xs:dateTime?
Variables		XV fn:adjust-dateTime-to-timezone (xs:dateTime? ,xs:dayTimeDuration?) : x:
Condition		XY fn:adjust-date-to-timezone (xs:date?) : xs:date? XY fn:adjust-date-to-timezone (xs:date?, xs:dayTimeDuration?) : xs:date?
Sort		
~~~	I	xy fn:adjust-time-to-timezone (xs:time? ,xs:dayTimeDuration? ) : xs:time?

18. Set the value of UserReturnCode to 0. Do this by using an Assign transform.

Expand the output assembly DBResp, and right-click UserReturnCode. Select Add Assign.

For each V	🖃 🔀 Employee	ResponseType		
🔏 Custom XPath 👻	⇒ <click filt<="" th="" to=""><th>er&gt;</th><th></th><th></th></click>	er>		
	🖃 📌 DBResp		[11] DBF	PereTure
[φ	e Userno	Undo Change o	[1 1] int	(i)
	e Row		oue	
	e Row	Redo		
		Revert		
	e Row	Cut		Ctrl+X
	e Row	Сору		Ctrl+C
	e SQL(			
	e squ	Paste		⊂trl+∀
		Delete		
	e sql			
		Open Informati		Ctrl+Shift+
	_	Open Declaratio	n	F3
	2	Cast		
	-	- Add Connection	ı	
		🕻 Accept All Auto		nsforms
		Reject All Auto		
		1	mappeu mar	ISTOFITIS
	4	Auto Map		
	_	Quick Link from	Input	Ctrl+L
	4	🕈 Add Assign		
		Database		
		Cache		

19. The default value for an Assign is "0", so leave this unchanged.

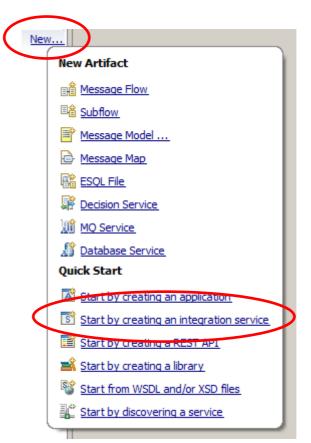


20. The submap will now look like this. The submap is complete, so save (Ctrl-S) and close the map.

getEmployees_subn	nap 🖉 🖓 💭 🖓 🕻	K & ☆  목 록  ¥  예 예  €		
글 🍪 ResultSet	[0*] Result Set Row	S Custom XPath ▼		
→I →I →I →I <click filter="" to=""></click>	2987 N. A. M.		🖃 📌 DBResp	[11] DBRespType
EMPNO	[11] CHAR	Assign 👻 🕕	<ul> <li>UserReturnCode</li> </ul>	[11] int
FIRSTNME	[11] VARCHAR		RowsRetrieved	[11] int
MIDINIT	[11] CHAR		e RowsAdded	[11] int
LASTNAME	[11] VARCHAR		e RowsUpdated	[11] int
WORKDEPT	[11] CHAR		e RowsDeleted	[11] int
PHONENO	[11] CHAR		e SQLCode_ErrorCode	[11] int
HIREDATE	[11] DATE		e SQLState_SQLState	[11] string
JOB	[11] CHAR		C SQL_Error_Message	[11] string
EDLEVEL	[11] SMALLINT			[0*] EMPLOYEETyp
- act	te et auro	-		

# 3. Create the Integration Service

1. In the Application Development navigator, click New, and "Start by creating an integration service".



2. Name the service EmployeeService, and accept the default to "Define it myself ...".

Since the schemas will be defined in a Shared Library, you should choose the option to store the interface definition in a shared library. Select this option, and specify the name of the EmployeeService_interface_and_maps.

Click Finish.

	new integration service ration service is an application with a well-defined interface. It implements flows for each service n.	2
Name	EmployeeService	
- Hame		
Descriptio	ion	
How will	vou define the interface for your integration service	
O Defir	ne it myself using the integration service editor	
C Imple	ement an interface already specified in an existing WSDL file	
	ement an interface already designed in Business Process Manager (.twx file)	
C Imple	ement an interface already designed in Business Process Manager (.twx file)	
C Imple	lement an interface already designed in Business Process Manager (.twx file) urce location of the existing WSDL file	
C Imple	ement an interface already designed in Business Process Manager (.twx file) arce location of the existing WSDL file ort an existing WSDL	
C Imple	lement an interface already designed in Business Process Manager (.twx file) urce location of the existing WSDL file	
C Imple	ement an interface already designed in Business Process Manager (.twx file)  rce location of the existing WSDL file ort an existing WSDL an existing WSDL in shared library	
C Imple The sour C Impo C Use	lement an interface already designed in Business Process Manager (.twx file)  arce location of the existing WSDL file ort an existing WSDL an existing WSDL in shared library  do you want to store your new interface definition	
C Imple The sour C Imple C Use Where d	lement an interface already designed in Business Process Manager (.twx file)  rce location of the existing WSDL file ort an existing WSDL an existing WSDL in shared library  do you want to store your new interface definition e in the Integration Service project	
C Imple The sour C Imple C Use Where c O Inline C In th	lement an interface already designed in Business Process Manager (.twx file) arce location of the existing WSDL file ort an existing WSDL an existing WSDL in shared library do you want to store your new interface definition ie in the Integration Service project the following shared library so that the service interface can access schema in the shared library.	
C Imple The sour C Imple C Use Where c O Inline C In th	lement an interface already designed in Business Process Manager (.twx file)  rce location of the existing WSDL file ort an existing WSDL an existing WSDL in shared library  do you want to store your new interface definition e in the Integration Service project	
C Imple The sour C Imple C Use Where c O Inline C In th	lement an interface already designed in Business Process Manager (.twx file) arce location of the existing WSDL file ort an existing WSDL an existing WSDL in shared library do you want to store your new interface definition ie in the Integration Service project the following shared library so that the service interface can access schema in the shared library.	
C Imple The sour C Imple C Use Where c O Inline C In th	lement an interface already designed in Business Process Manager (.twx file) arce location of the existing WSDL file ort an existing WSDL an existing WSDL in shared library do you want to store your new interface definition ie in the Integration Service project the following shared library so that the service interface can access schema in the shared library.	
C Imple The sour C Imple C Use Where c O Inline C In th	lement an interface already designed in Business Process Manager (.twx file) arce location of the existing WSDL file ort an existing WSDL an existing WSDL in shared library do you want to store your new interface definition ie in the Integration Service project the following shared library so that the service interface can access schema in the shared library.	

3. The following service editor will open and the following service interface will be generated. The service will contain a request/response operation named operation1.

Note that the location of the service interface is EmployeeService_interface_and_maps, with the name of the wsdl = EmployeeService.wsdl.

EmployeeSer	vice 🛛 📃		
•Interface			
Name	EmployeeServio	ie in the second s	
Namespace	http://Employe	eService	
Location	/Employee Corry		
Location	/Employeeservi	ice_interface_and_maps/EmployeeService.wsdl	
•Operations			
•Operations	d their paramete		Туре
• Operations	d their paramete	≱ 🖻   🔊 🔊 🔊 🖉   💬	Туре
• Operations Operations an Message Typ	d their paramete	≱ 🖻   🔊 🔊 🔊 🖉   💬	Type string

4. Note that the Library Reference of the new service has automatically been set to reference the Library where the service schemas and will be defined (right-click the new Integration Service and select Manage Library References).

After reviewing, cancel this window.

🌐 Manage Library References	×
Select the shared or static libraries to be referenced. Any other static libraries referenced from the selected static library will also be included	: 1.
Shared libraries	
EmployeeService_interface_and_maps	
Static libraries	
, The following static libraries will also be included because they are referenced by the selected static libraries.	
-	
OK Cancel	

- 5. Change the name of operation1 to getEmployee:
  - Highlight operation1, and overtype with getEmployee
  - Press the Return key after renaming, which will automatically allocate the names of the message types for the operation

Configuration Name	EmployeeServ	ice	
Namespace	http://Employe	eeService	
Location	/EmployeeSer	vice_interface_and_maps/EmployeeService.wsdl	
	nd their paramet		_
Operations ar Message Ty	-	Name	Туре
Message Ty	pe nployee	Name	 
Message Ty getEn Di getEmpl	pe nployee		Type string string

6. Change the Type of the input message to **employeeNumber**. (Click the current value "string"), then click Browse).

T	ype	
S	tring	
	Browse	
	New	
	🖃 boolean	
	🖃 date	
	🖃 dateTime	
	🖃 double	
	🖃 float	
	🖃 hexBinary	
	🖃 int	
	🖃 string	-
	•	

- 7. Type "e" into the filter, and select employeeNumber.
  - Click OK.

Data Type Selection		
Filter by type, namespace, or file (	? = any character, * = a	any string):
e		New
Filtered data types:		
C EMPLOYEE		
employeeNumber		
= employeeNumberType		
EmployeeResponse		
EmployeeResponseType		
EMPLOYEEType		
= encodingStyle		
= ENTITIES		
ENTITY		
e Envelone		
Qualifier:		
e http://www.example.org/emp	oloveeNumber - Employee	eService s
	, , , , ,	_
<u></u>		
?	ОК	Cancel

8. For the output message, use the same technique and select **EmployeeResponse**.

Data Type Selection		<u>- 🗆 ×</u>
Filter by type, namespace, or file	(? = any character, * =	any string):
e		New
Filtered data types:		
C EMPLOYEE		
e employeeNumber		
employeeNumberType		
EmployeeResponse		
EmployeeResponseType		
EMPLOYEEType		
🖃 encodingStyle		
ENTITIES		-
Qualifier:		_
€ http://hrdb/iibadmin - Employ ◀	eeService_submaps/hrd	lb/iibadmin/ŀ ▶
?	ОК	Cancel

9. The getEmployee operation interface is now fully defined, and should look like this.

Note that the name of the input message will have been changed to employeeNumber. This is because employeeNumber is an XSD element, and the "name" attribute is not permitted when using an XSD element.

Similarly, for the output message, you have used an element (EmployeeResponse). You could have used the element type (EmployeeResponseType), which would not have changed the name of the operation types.

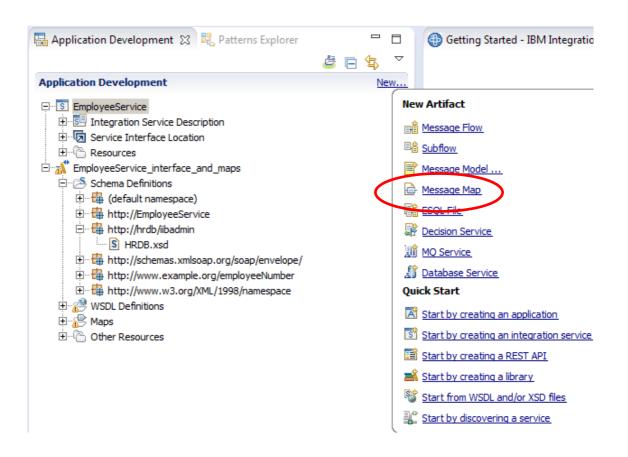
Save the service at this point (Ctrl-S).

📝 *EmployeeSe	ervice 🛛		
▼Interface			
Configuration			
Name	EmployeeServio	e	
Namespace	http://Employe	eService	
Location	/EmployeeServi	ce_interface_and_maps/EmployeeService.wsdl	
•Operations ar	s 🐉 🖏		
Message Ty	pe	Name	Туре
🔻 👹 getEm	ployee		
D getEmpl	oyee	employeeNumber	employeeNumber
🕼 getEmpl	oyeeResponse	EmployeeResponse	EmployeeResponse
Service 🔳	Interface		

### 3.1 Implement the Main Map

You now need to create the main map that will invoke the submap. The main map will be aware of the type of protocol that invokes it (in this case an integration service). It will extract the payload of the message (employeeNumber), and pass this to the submap to access the database. The main map will be located in the shared library EmployeeService_interface_and_maps.

1. Highlight the EmployeeService_interface_and_maps library, and click New, Message Map.



2. Ensure the container is correctly identified.

Name the new map getEmployee_WS

Click Next.

🌐 New Message Map	
Specify a new message map file Select map type, container, name, and broker schema for the new map.	
Type of map that you want to create: Message map called by a message flow node Submap called by another map	
Container: EmployeeService_interface_and_maps	New
Map name: getEmployee_WS	
Map organization Vise default broker schema Schema: (default broker schema)	
Output Search Next > Finish	Cancel

- 3. For both the map input and output, expand EmployeeService_interface_and_maps, and expand DFDL and XML Schemas.
  - For the input, select **getEmployee** (remember this originates from the input message of the integration service, and resolves to element employeeNumber).
  - For the output, select **getEmployeeResponse** (resolves to element EmployeeResponse).

Click Finish.

Wew Message Map Select map inputs and outputs Creates a map that can contain message inputs and outputs with the Properties for 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):
J Select map inputs	Select map outputs
** EmployeeService_interface_and_maps         ** EmployeeService_interface_and_maps         ** EmployeeService_interface_and_maps         ** EmployeeResponse {http://hrdb/ibadmin}         ** EmployeeResponse {http://bendoyeeService}         ** Bettemployee {http://schemas.xmlsoap.org/soap/envelope/}         ** Bettemployee {http://schemas.xmlsoap.org/soap/envelope/}         ** BettemployeeResponse {http://EmployeeService}         ** BettemployeeResponse {http://EmployeeService}         ** BettemployeeResponse {http://EmployeeService}         ** BettemployeeResponse {http://EmployeeService}         ** BettemployeeService_interface_and_maps         ** BettemployeeService_Interface_and_maps         ** BettemployeeService}         ** EmployeeService         ** BettemployeeService}	Mean Provide a service interface and maps     DFDL and XML Schemas     DFDL and XML Schemas     DFDL and XML Schemas.xmlsoap.org/soap/envelope/}     DBResp {     DPARTMENT {http://hrdb/ibadmin}     DEPARTMENT {http:/hrdb/ibadmin}     DEPARTMENT {http://hrdb/ibadmin
?	< Back Next > Finish Cancel

4. The map editor will open. Expand both the input and output assemblies.

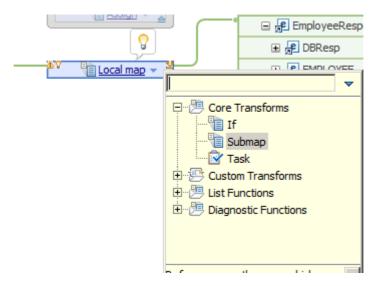
getEmployee_WS				
getEmployee_WS	🗶   4 rd   🐔 🐔 🕸 r4	🖙 🛱   🔞 🕲 🐌   🎟 🖮 📾 🕯	i 📰 🛛 📴	
Message Assembly	getEmployee		🖃 🔩 Message Assembly	getEmploy
⇒ <click filter="" to=""></click>			⇒ <click filter="" to=""></click>	
		Overrides	🗄 📌 Properties	[01] Properties
🗉 📌 Properties	[01] PropertiesType		🖃 📌 getEmployeeResponse	[11] <anonymo< td=""></anonymo<>
			🖃 📌 EmployeeResponse	[11] EmployeeR
🖃 📌 getEmployee	[11] <anonymous></anonymous>	-	🗄 📌 DBResp	[11] DBRespTyp
employeeNumber	[11] employeeNumberType	-	🗉 📌 EMPLOYEE	[0*] EMPLOYEE

5. Connect the input **employeeNumber** to the output **EmployeeResponse**. This will produce a Local Map.

A warning will indicate that there are no valid element mappings. We will resolve this by changing the Local Map to a Submap.

getEmployee_WS	P1 😰	🗙   🕸 🕼 🞼 🞼	두 음  🔞 🐌 🐌 🔳 📾 🏭 📾 📰	
☐ [™] Message Assembly → <click filter="" to=""></click>		getEmployee	□ Lag Message Assembly	getEmploy
€ 📌 Properties	[01]	PropertiesType	Overrides       Image: Move →       Image: Move → <td>[01] Properties [11] <anonymo [11] EmployeeF</anonymo </td>	[01] Properties [11] <anonymo [11] EmployeeF</anonymo 
🖃 📌 getEmployee	[11]	<anonymous></anonymous>	⊕ 📌 DBResp	[11] DBRespTy
📌 employeeNumber	[11]	employeeNumberType	Local map -	[0*] EMPLOYEE

6. Change the Local Map to a Submap (click on the blue arrow).



The submap will show an error, since you haven't yet provided the name of the submap.

	🛨 📌 DBResp
Submap - O	🗉 📌 EMPLOYEE
<b>! Error</b> The Submap transformation must refer t	to another map.

7. Highlight the submap transform, and locate the Properties of the transform.

	[11] <anonymous></anonymous>	
r	r [11] employeeNumberType	Submap 👻 🕀 😥
ems	ems 🚦 Outline 🧔 Tasks 🖽 Deployment Log	
ge !	getEmployee_WS.map	Browse New

8. Click Browse, and select the **getEmployees**.submap located in the EmployeeService_interface_and_maps library. Click OK.

Note - the map selection wizard will initially only show maps or submaps that are available for use in the current context. The screen capture below has unchecked this option, enabling you to see all maps and submaps, even though not all of these choices would make sense in the current context.

Submap Selection	
Choose a resource:	
Pattern (? = any character, * = any string):	
Matching Resources:	
🔁 getEmployee_WS.map	
getEmployees_submap.map	
T. F. Harris	
EmployeeService_interface_and_maps	
Show only applicable maps	
🤉 ок са	ancel
	Choose a resource: Pattern (? = any character, * = any string): Matching Resources: © getEmployee_WS.map © getEmployees_submap.map © getEmployees_submap.map In Folders: © EmployeeService_interface_and_maps © Show only applicable maps

Integration Service using maps and submaps

9. The main map is now complete, so save (Ctrl-S) and close.

*getEmployee_V	vs 🖂				
🗟 getEmployee_W	/S				
<b>▼getEmployee_</b>	ws	at (a)   🍕 🗳 🕼	×  🗧 🎝 🐧 🐌 🐌 💷 📾 📾 🕮 🗉		
⊟ Bessage A ⇒ <click filte<="" p="" to=""></click>		getEmployee	E La Message Assemi → <click filter="" to=""></click>		
الله الله الله الله الله الله الله الل	oyee	[01] PropertiesType [11] <anonymous> rr [11] employeeNumberType</anonymous>	Overides Move → Submap → Submap → Overides B gP Properties B gP petEmployeeRe B gP EmployeeRe B gP EmployeeRe B gP EmployeeRe B gP EmployeeRe B gP EmployeeRe B gP EmployeeRe B gP EmployeeRe	sponse [11] EmployeeRes; [11] DBRespType	
Transform - Submap					
General	File:	getEmployees_submap.map	Browse New		
Cardinality	Map:	getEmployees_submap			
Variables					
Condition					

### 3.2 Implement the getEmployee Operation

You now need to complete the implementation of EmployeeService, so return to the service editor.

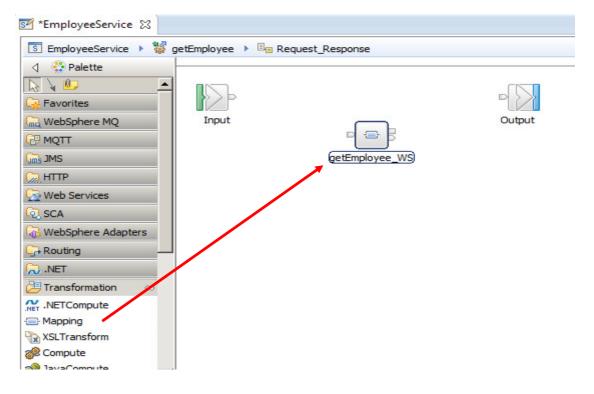
1. In the EmployeeService, click the Service tab.

You will see that the getEmployee operation is not implemented, so is greyed out. Click getEmployee to implement it.

🕞 Application 😢 🍀 Patterns Expl 😑 🗖	🔁 getEmployee_WS 🛛 🖾 EmployeeService 😒
5 E \$ V	S EmployeeService >
Application Development New	
EmployeeService     Integration Service Description     Service Interface Location     Resources     Prove Flows     Prove Subflows     Subflows     Referenced Libraries	
EmployeeService_interface_and_maps     Schema Definitions     Gefault namespace)     Gefault namespace)     Gefault namespace)     Gefault namespace)     Gefault namespace)     Gefault namespace)     Gefault namespace     Gefault namespa	Era Catch
G getEmployee_WS.map	Service 1 Interface

2. From the Transformation folder, drop a Mapping Node onto the flow editor.

Name it getEmployee_WS.



3. In the Properties (Basic) of the new Mapping Node, click Browse to select the map you created earlier. Select the getEmployee_WS map in the Shared Library, and click OK.

*EmployeeService 🔯	- 8
🗄 EmployeeService 🕨 👹 getEmployee 🕨 📴 Request_Response	• H
1 🎡 Palette	
Favorites	
WebSphere MQ Input Output	
ΜQTT	
ata Transformation Map Selection	
HTTP Filter names (? = any character, * = any String):	
Web Services	
SCA Select a Data Transformation Map	
WebSphere Adapters	ico interfaco and
Routing	ce_interface_and
).NET	
Transformation 🗱	
.NETCompute	
) Mapping Location:	
XSI Transform  aph User Defined Properties	
Service 1 Interface	
Properties 🕱 🎇 Problems 🔡 Outline 🖉 Tasks 🏢 Deployment Log	
OK OK	Cancel
> Mapping Node Properties - getEmployee_WS	
escription	$\frown$
asic Mapping routine* {gen}:getEmployee_Request_Response_getEmployee_WS	Browse
ilidation Transaction* Automatic	

4. The new map node properties will show the map in the Shared Library.

Properties S	🕱 🖳 Problems	🗄 Outline 🖉 Tasks 🔠 Deployment Log
🖶 Mapping	Node Propertie	s - getEmployee_WS
Description		
Basic	Mapping routine*	{default}:getEmployee_WS in Shared Library EmployeeService_interface_and_maps
Validation	Transaction*	Automatic
Monitoring	Transacuor	Automauc

5. Connect the nodes as shown.

Input (getEmployee_WS)

Save the flow.

## 4. Test the Integration Service with the Flow Exerciser

The Flow Exerciser is a new function in IIB v10. You will use this to perform a simple unit test of the service you have just developed. You will see additional Flow Exerciser functions in later labs.

First, make sure that the node TESTNODE_iibuser is started. You can check this in the Toolkit Integration Nodes pane. If it isn't started, right-click and select Start.

1. In the service editor, make sure the Service tab is selected. You will see a red button in the top right of the editor pane.

EmployeeService 🔀		
S EmployeeService >		• 🗗
S EmployeeService	(I) EmployeeService	
SOAP/HTTP Binding	10 getEmployee	
	🔀 Error Handlers	
	E Failure	
	B Catch	
	B _☉ <u>Timeout</u>	
	· · · · · · · · · · · · · · · · · · ·	-
Service 🛈 Interface		
🔲 Properties 🛛 🔝 Problems 🗄 Out	tine 🖉 Tasks 🌐 Deployment Log	≝ ▽□□

2. Click the red button. The integration service will be deployed to the TESTNODE/default. Note that if you have other nodes or servers running, you will be presented with a selection dialogue.

When the service is deployed, you will see an information message. After reading this, select the box to not see in the future, and click Close.

Ready to record message	×
a Ready to record message	
Integration server [server 1] on integration node [IB10NODE] is ready to record messages. You can send a message to the flow by using one of the following actions:	
<ul> <li>Click Send message icon to create an input message, or select an existing message and send it to the flow.</li> </ul>	
<ul> <li>Use a third party tool or client to send an input message to the flow. After the message is processed by the flow, you must click <b>Show path</b> to highlight the message path.</li> </ul>	
You can click a highlighted connection on the flow to see the details of the message that passed through the connection. If you click the first highlighted connection, you can save the recorded message for future use.	
Do not show this message again	
Close	

3. The flow editor will turn grey, indicating that the Flow Exerciser is ready for use.

In the top right, the available icons will have changed. Select the "Send message" icon (the middle icon).

EmployeeService 🛛		
S EmployeeService >		🛛 🖗 🕅
S EmployeeService	EmployeeService     getEmployee	
	Error Handlers En Failure En Catch	
		1

4. In the Send Message window, click the button to create a new message.

🌐 Send Message	
Send Message Create or select a message to send to the flo	w. Click the message category header (e.g. Input Messages) for more information.
Input Messages	Manage your messages: 2 - Click 'New' to create an input message. 2 - Click 'Duplicate' to copy the selected input message. 2 - Click 'Delete' to remove the selected messages. - Click a message to view or edit it.

5. Name the new message "Employee 000010".

The Flow Exerciser will have automatically populated a template input message, based on the WSDL and associated service operation input message.

In the employeeNumber element, change the value to 000010, and click Send.

Send Message	
end Message	
Create or select a message to send to t	he flow. Click the message category header (e.g. Input Messages) for more information.
zi 🛜 🗙	
	Name Employee 000010
Employee 000010	Main
	Input Location: SOAP Input
	Soap operation getEmployee
	Message Details
	Edit, type, or import a message.
	<pre><?xml version = "1.0" encoding = "UTF-8?><tns0:envelope <tns0:header="" xmlns:tns0="http://scher."></tns0:envelope> <tns0:body> <tns1:getemployee> <employeenumber>employeeNumber</employeenumber> </tns1:getemployee>  </tns0:body></pre>
	Show in hexadecimal viewer (Read Only)
	Export Source
	Apply Revert
2	Send Close

6. The service operation will be executed. Highlighting the "Sending Message" line will show the message that was sent to the service.

Progress Information	_ <b>_</b> X
Invoke Message Flow (Employee 000010)     Message flows deployment successfully completed     Starting     Sending Message to "SOAP Input"     Received HTTP reply message for "SOAP Input"     Stopped	
<pre><?xml version="1.0" encoding="UTF-8"?><tns0:envelope xmlns:q10="http://www.example.org/employeeNumber" xmlns:tns0="http://schemas.xmlsoap.org/soap/envelope/" xmlns:tns1="http://EmployeeService" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"></tns0:envelope></pre>	

Integration Service using maps and submaps

Version 10.0.0.3

7. Highlighting the "Received HTTP" line will show the message returned from the service. Note that the employee record for the selected employee number has been retrieved from the EMPLOYEE table.

Also note the UserReturnCode = 0, and RowsRetrieved = 1.

Progress Information	
Invoke Message Flow (Employee 000010)     Message flows deployment successfully completed     Starting     Sending Message to "SOAP Input"	
Received HTTP reply message for "SOAP Input"	
xml version="1.0" encoding="UTF-8"? <soapenv:envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"&gt;</soapenv:envelope 	
<pre><soapenv:body> <io2:getemployeeresponse xmlns:io="http://hrdb/iibadmin" xmlns:io2="http://EmployeeService" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"> <io:employeeresponse> <io:employeeresponse> </io:employeeresponse></io:employeeresponse></io2:getemployeeresponse></soapenv:body></pre>	
<ul> <li><userreturncode>0</userreturncode></li> <li><rowsretrieved>1</rowsretrieved></li> <li><rowsadded xsi:nil="true"></rowsadded></li> <li><rowsupdated xsi:nil="true"></rowsupdated></li> <li><rowsdeleted xsi:nil="true"></rowsdeleted></li> <li><solcode errorcode="" xsi:nil="true"></solcode></li> </ul>	
<sqlstate_sqlstate xsi:nil="true"></sqlstate_sqlstate> <sql_error_message xsi:nil="true"></sql_error_message>  <out:employee xmlns:out="http://hrdb/iibadmin"> <empno>000010</empno></out:employee>	
<firstnme>CHRISTINE</firstnme> <midinit>I</midinit> <lastname>HAAS</lastname> <workdept>A00</workdept> <phoneno>3978</phoneno> <hiredate>1995-01-01</hiredate>	<b>T</b>
	Close

Close the Progress Information window.

8. In the flow editor, you will now see that some parts of the service definition have turned green. Click the "getEmployee" operation name.

EmployeeService ☎	
S EmployeeService >	
S EmployeeService	
SOAP/HTTP Binding	detEmployee
	K Error Handlers
	[□] _□ <u>Failure</u>
	E Catch
	□ _□ <u>Timeout</u>

9. In the flow editor, you will see the subflow that implements this operation. You will see that the connectors are green, indicating that the message flowed down this path.

🗵 EmployeeService 🕨 👹 g	getEmployee 🕨 🗏 🗉 Request_Response
👌 😳 Palette	
📐 🖌 🕼 🛌	
🙀 Favorites	
🖳 WebSphere MQ	Input M Output
🖓 MQTT	
Gms JMS	getEmployee_WS
💭 HTTP	
🚱 Web Services	
🔁 SCA	
🐻 WebSphere Adapters	
Consultant	

10. Click the icon on the first connector.

The message tree that was current at the time it passed through this connector will be retrieved and displayed. Note that this is the full message tree, not just the user data.

You can expand various parts of the tree, such as LocalEnvironment, Environment, ExceptionList, as well as the Message.

Note that the domain of the message tree is shown. In this case, it is XMLNSC.

Input	Output
	Recorded Message
	Environment
	Local Environment
	Exception List
	▼ Message
	<pre>_ <message></message></pre>
	⊟ <xmlnsc></xmlnsc>
	<xmldeclaration></xmldeclaration>
	<version>1.0</version>
	<encoding>UTF-8</encoding>
	NS1:getEmployee>
	<employeenumber>000010</employeenumber>

11. Click the icon on the final connector. This time, you will see the message tree after the employee record has been retrieved from the database.

Note that you can return to the first connector icon to review the message tree again, at that point. The data is not discarded.

	Output
getEmployee_WS	Recorded Message
	Environment
	Local Environment
	Exception List
	▼ Message
	□ <message></message>
	io2:getEmployeeResponse>
	io:EmployeeResponse>
	<out:employee> <empno>000010</empno></out:employee>
	<firstnme>CHRISTINE</firstnme>
	<midlinit>i</midlinit>
	<lastname>HAAS</lastname>
	<workdept>A00</workdept>
	<phoneno>3978</phoneno>
	<hiredate>1995-01-01</hiredate>
	<job>PRES </job> <edlevel>18</edlevel>
	<ex.>F</ex.>

12. Click the Send Message icon again.



13. Create a new test as before. Name it "Employees matching 0020".

Set the employeeNumber to 0020, and click Send.

赵 🏧 🗶	
□	Name       Employees matching 0020         Main       Input Location:       SOAP Input         Soap operation       getEmployee
	Show in hexadecimal viewer (Read Only)
	Export Source

14. The returned message will contain two employees, both containing "0020" in the employee number field.

RowsRetrieved = 2.

Progress Information	<u>_ 0 ×</u>
⊡…E Invoke Message Flow (Employees matching 0020)	
Message flows deployment successfully completed	
🗄 🖓 Starting	
Sending Message to "SOAP Input"	
Received HTTP reply message for "SOAP Input"	
Stopped	
xml version="1.0" encoding="UTF-8"? <soapenv:envelope< td=""><td><b></b></td></soapenv:envelope<>	<b></b>
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">	
<soapenv:body></soapenv:body>	
<io2:getemployeeresponse <="" td="" xmlns:io="http://hrdb/iibadmin"><td></td></io2:getemployeeresponse>	
xmlns:io2="http://EmployeeService">	
<out:employeeresponse xmlns:out="http://hrdb/iibadmin"></out:employeeresponse>	
<dbresp> <userreturncode>0</userreturncode></dbresp>	
<rowsretrieved>2</rowsretrieved>	
<out:employee></out:employee>	
<empno>000020</empno>	
<firstnme>MICHAEL</firstnme>	
<midinit>L</midinit>	
<lastname>THOMPSON</lastname>	
<workdept>801</workdept>	
<phoneno>3476</phoneno>	
<pre><hiredate>2003-10-10</hiredate> <job>MANAGER </job></pre>	
<edlevel>18</edlevel>	
<sex>M</sex>	
<birthdate>1978-02-02</birthdate>	
<salary>94250</salary>	
<bonus>800</bonus>	
<comm>3300</comm>	
<out:employee></out:employee>	
<empno>000200</empno> <firstnme>DAVID</firstnme>	
<pre></pre>	
<lastname>BROWN</lastname>	
<workdept>D11</workdept>	
<phoneno>4501</phoneno>	
<hiredate>2002-03-03</hiredate>	
<job>DESIGNER </job>	
<edlevel>16</edlevel>	
<sex>M</sex>	_
	<u> </u>
	diam
	Close

15. Create a new test as before. Name it "Employee 000012".

This time, select "Import from file". Using the File system button, navigate to c:\student10\integration_service\data\employeeNumber 000012.xml.

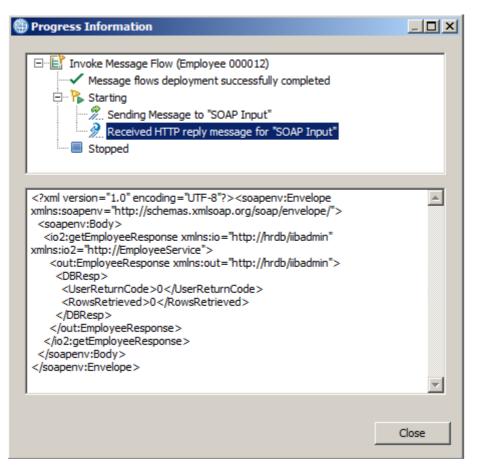
Name Employee 000012	
Input Location: SOAP Input Soap operation getEmployee Message Details	▲
Edit, type, or import a message.	Workspace File system

16. Click Send, and use the same tools to review the test outcome.

🌐 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.
Input Messages     Employee 000010     Employees matching 0020     new message 1	Name       Employee 000012         Main       Input Location:       SOAP Input         Soap operation       getEmployee         Message Details       Edit, type, or inport a message.         ✓       Import from file         File name:       C:\student10\Integration_service\data\employeeNumber 000012.xml       Workspace <pre>       </pre>
?	Send Close

17. You will see no rows have been returned, and the EMPLOYEE part of the message is not present.

RowsRetrieved = 0.



## 5. Test the service with SOAPUI

1. In the IIB Toolkit, select the EmployeeService in the node TESTNODE_iibuser, and select the Properties.

Highlight the Service Query URL, and right-click and select Copy.

‰L. ⊠ ‰L. ‱D ∭D □ □	🔲 Properties 🕱 🔝 Problems 🚦	Outline 🖉 Tasks 🏢 Deployment Log	
	Property	Value	
🖃 📲 Integration Nodes	⊡ Info		
IB10NODE	Deployment Time	Wed Dec 02 16:47:45 GMT 2015	
TESTNODE_iibadmin	Full Name	EmployeeService.appzip	
E TESTNODE_iibuser	Last Modified	Wed Dec 02 16:47:44 GMT 2015	
🚊 default	Operations	getEmployee	
EmployeeService	Service Query URL	http://192.168.126.132:7800/EmployeeService?wsd	
EmployeeService_interface_and	Service URL	http://192.168.126.132:7800/EmployeeService	
EmployeeService_internet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enternet_enter	<ul> <li>Keywords</li> </ul>	Restore Default Value	
En proyector vice_submaps	BAR	C:/Users/iibuser/IBM/IIBT10/workspace2/TestModeBar	
	VERSION		

2. Open SOAPUI (on the Windows Start menu). Ensure you are using the SOAPUI workspace **EmployeeService_PrebuiltWorkspace**.

Create a new SOAPUI project (File, New SOAP Project).

Provide a suitable name of your choosing for the project (not one of the existing projects).

In the "Initial WSDL" field, paste the contents of the Windows clipboard.

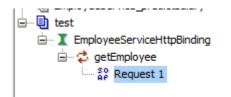
Important - the Copy function included the text "Service Query URL" at the start of the string, so this must be manually deleted.

Click OK.

🔶 New SOAP Pro	ject	×
New SOAP Pro	<b>ject</b> L/SOAP based Project in this workspace	
Project Name:	test	
Initial WSDL:	http://192.168.126.132:7800/EmployeeService?w Browse	
Create Requests:	Create sample requests for all operations?	
Create TestSuite:	Creates a TestSuite for the imported WSDL	
Relative Paths:	Stores all file paths in project relatively to project file (requires save)	
		OK Cancel

3. Expand the new project, and open Request1. You will see that SOAPUI has pre-populated the web service input with the required schema elements. In this case, these are derived from the EMPLOYEE table.

Also note that the service port number has been set to 7800. This is the port number that has been allocated for the TESTNODE default server.



4. Provide a value for the EMPNO element. We suggest using value 0020 (replace the ? with this value, as shown).



5. Clicking the green arrow (top left), and the Integration Service should run and return the data to the client, similar to this, depending on the number of rows retrieved.



6. Create a second SOAPUI request in the same project. This time, use the value 000012, which does not exist in the EMPLOYEE table.

When you click the green arrow, you should see the response indicating that the flow successfully accessed the tables (user return code 0), and no rows were returned (RowsRetrieved = 0).

```
(Ę
  Soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  Ð
       <soapenv:Body>
Raw
  <io2:getEmployeeResponse xmlns:io2="http://EmployeeService" xmlns:io="http
   \Box
             <io:EmployeeResponse>
                <DBResp>
                   <UserReturnCode>0</UserReturnCode>
                   <RowsRetrieved>0</RowsRetrieved>
                   <RowsAdded xsi:nil="true"/>
                   <RowsUpdated xsi:nil="true"/>
                   <RowsDeleted xsi:nil="true"/>
                   <SQLCode_ErrorCode xsi:nil="true"/>
                   <SOLState_SOLState xsi:nil="true"/>
                   <SQL Error Message xsi:nil="true"/>
                </DBResp>
             </io:EmployeeResponse>
          </io2:getEmployeeResponse>
       </soapenv:Body>
    </soapenv:Envelope>
```

## END OF LAB GUIDE