

## IBM Integration Bus

# Creating an Integration Service accessing Microsoft® SQL Server

### Featuring:

- Basic configuration of SQL Server
- Defining a new SQL Server data and table
- Setting SQL Server permissions for IIB
- Configuring IIB node for SQL Server
- Testing IIB application with the Flow Exerciser

**April 2016**

Hands-on lab built at product  
Version 10.0.0.4

|  |           |
|--|-----------|
| <b>1. INTRODUCTION .....</b>   | <b>3</b>  |
| 1.1 SCENARIO OVERVIEW .....  | 3         |
| 1.2 OUTLINE OF TASKS.....  | 4         |
| <b>2. SQL SERVER CONFIGURATION.....</b>                              | <b>5</b>  |
| 2.1 CONFIGURE SQL SERVER NETWORK.....                                | 5         |
| 2.2 CREATE HRDB DATABASE .....                                       | 7         |
| 2.3 CONFIGURE SECURITY .....   | 12        |
| 2.3.1 <i>Create login credentials for iibuser</i> .....              | 12        |
| 2.3.2 <i>Define database authorisation rules for HRDB</i> .....      | 14        |
| 2.3.3 <i>Validate database schema and table permissions</i> .....    | 16        |
| 2.3.4 <i>Check access from iibuser</i> .....                         | 18        |
| <b>3. CREATE AND CONFIGURE THE IIB NODE AND APPLICATION .....</b>    | <b>20</b> |
| 3.1 CREATE IIB NODE .....  | 20        |
| 3.2 SET WINDOWS PATH FOR SQL SERVER AUTHORISATION FOR IIB.....       | 24        |
| 3.3 CREATE DATABASE DEFINITION FILE FOR HRDB .....                   | 25        |
| 3.4 IMPORT THE PREBUILT INTEGRATION SERVICE.....                     | 32        |
| <b>4. TEST THE INTEGRATION SERVICE WITH THE FLOW EXERCISER .....</b> | <b>33</b> |
| <b>END OF LAB GUIDE .....</b>  | <b>44</b> |

*Microsoft, Windows and SQL Server are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.*

# 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.

This lab is based on Lab 1 in this series, Creating an Integration Service. Lab 1 uses a DB2 database, and the scenario creates an integration service that retrieves rows from the EMPLOYEE table in the HRDB database. This is done by using a map to extract the request from the incoming web service request. The map invokes a submap which accesses the database.

For more details, please review the lab guide.

## 1.1 Scenario Overview

In this lab, you will make the necessary configurations in SQL Server and IIB to enable an IIB application (integration service) to connect to SQL Server, and retrieve some rows from a specific SQL Server table.

The integration service is provided for you, and is the solution of the integration service that was developed in Lab 1 (Create Integration Service) in this series of labs.

## 1.2 Outline of tasks

The tasks to complete in this lab are the following:

1. Configure port in SQL Server for TCPIP clients
2. Create a new database in SQL Server.
3. Configure the appropriate permissions for the database.
4. Configure the appropriate Windows configuration for the iibuser userID.
5. Create a new IIB node
  - a. Specify the required JDBC configurable service
6. Use the IIB Toolkit to create a new database definition
7. Import the existing solution for the integration service, EmployeeService
8. Test the integration service using SQL Server

## 2. SQL Server Configuration

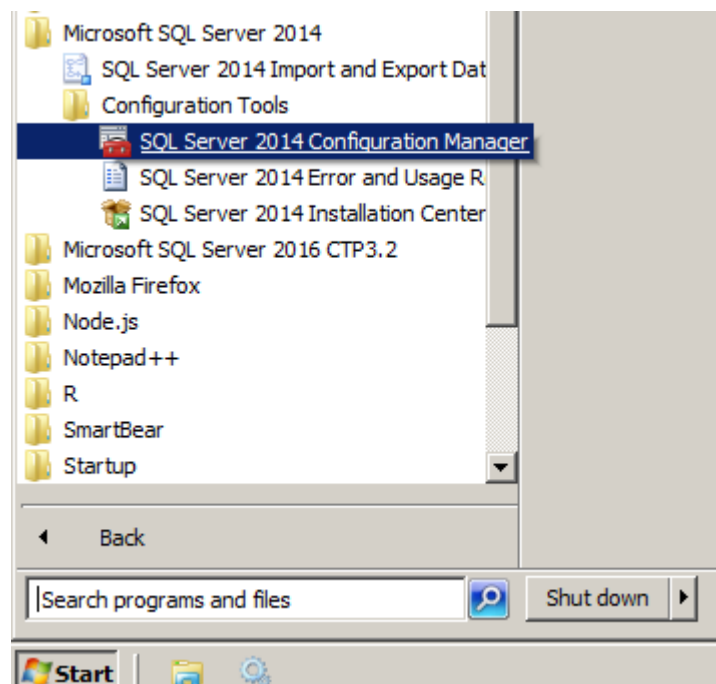
### 2.1 Configure SQL Server Network

The application that you will use in this lab will use a database named HRDB. This is based on the same HRDB database that was created for DB2 in Lab 1.

In this lab, the HRDB database will be created by the **iibadmin** user, and permissions will be given to the user **iibuser**, so that the IIB node TESTNODE\_iibuser will be able to access the database.

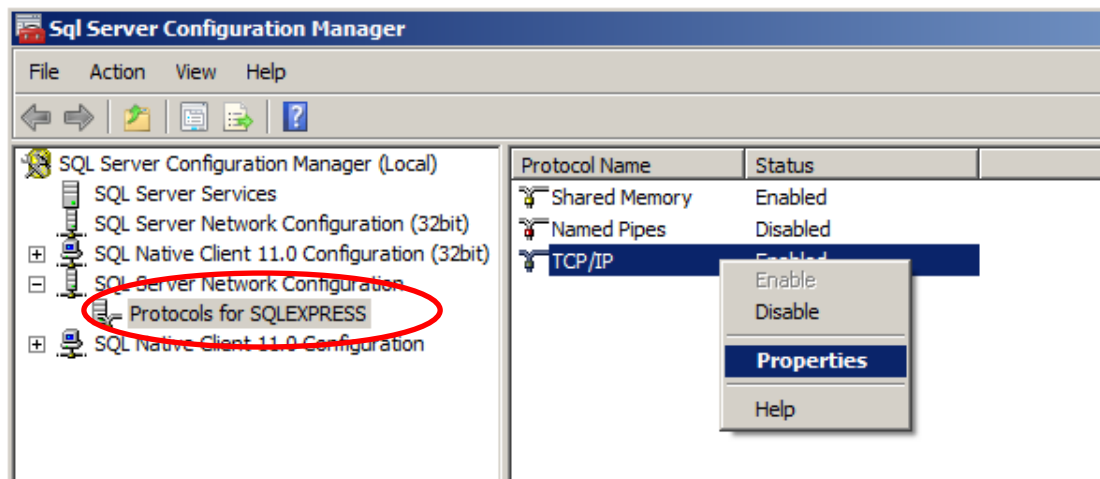
1. Login to Windows with the user **iibadmin**, password = **passw0rd**.
2. From the Windows Start menu, open :

**Microsoft SQL Server 2014  
Configuration Tools  
SQL Server 2014 Configuration Manager**

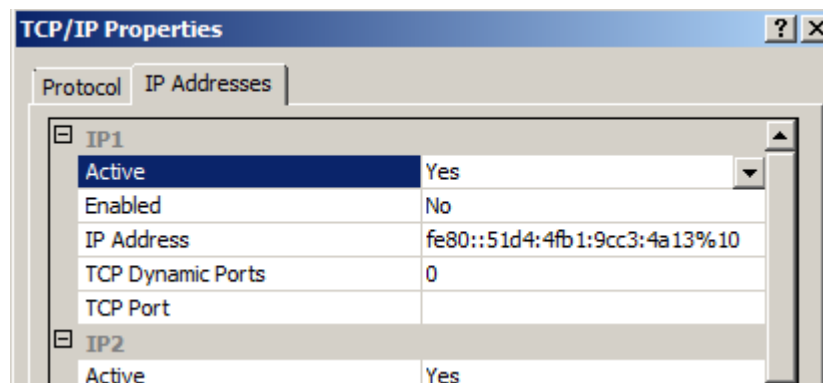


- Expand SQL Server Network Configuration, and select Protocols for SQLExpress.

Right-click the "TCP/IP" protocol, and select Properties.



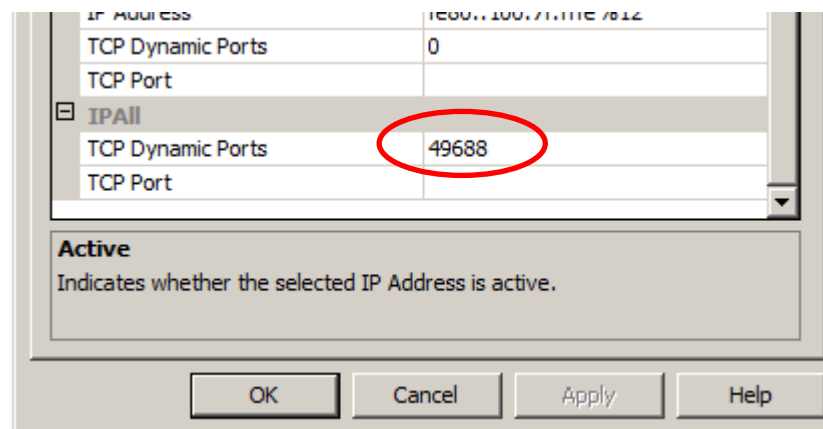
- Select the "IP Addresses" tab.



- Go to the bottom of the page.

Set TCP Dynamic Ports" to 49688. This port can be varied - we have just picked this at random. It will be referenced by the SQL Server Client, and by the IIB node, so please use this value in the workshop scenario.

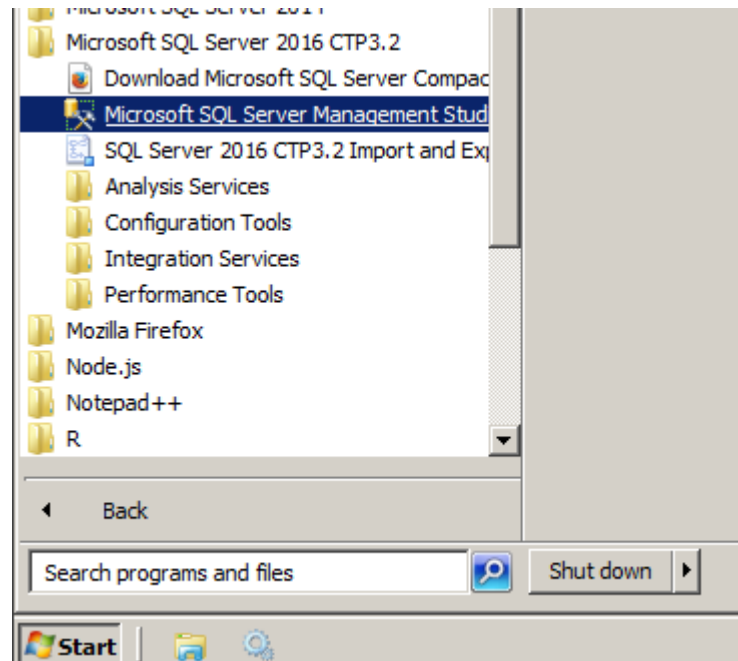
Click OK.



- Close SQL Server Configuration Manager

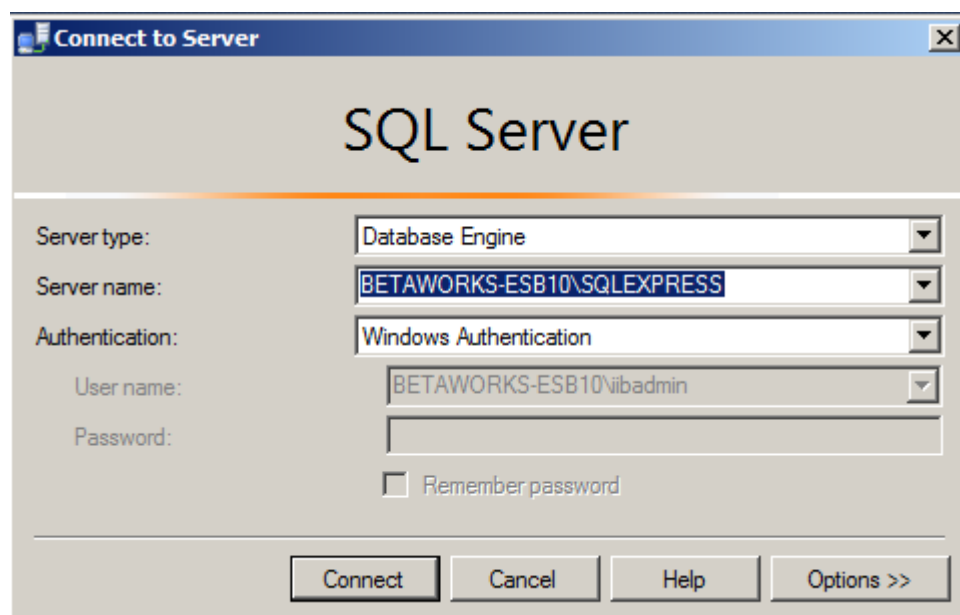
## 2.2 Create HRDB database

1. Again with the **iibadmin** user, from the Start menu, expand Microsoft SQL Server 2016 CTP 3.2, and select Microsoft SQL Server Management Studio.

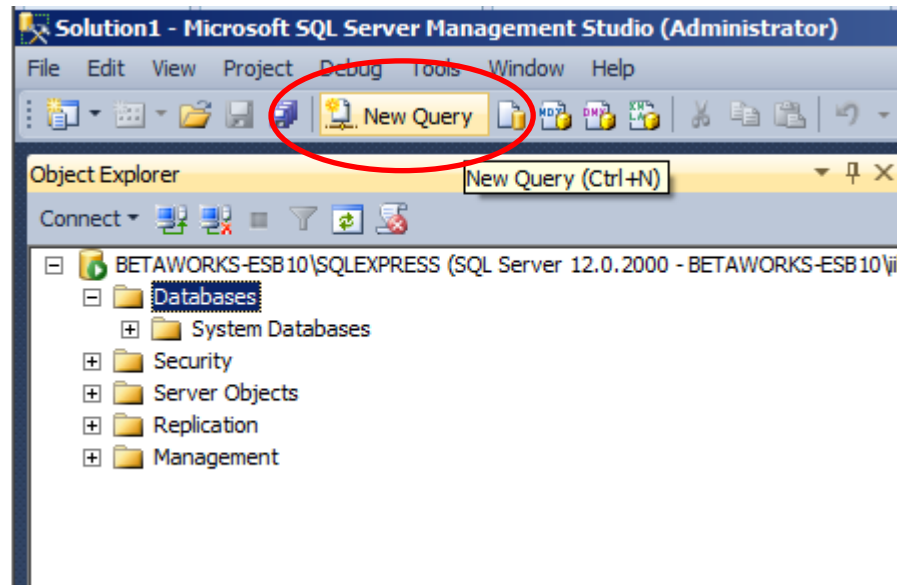


2. Ensure Authentication is set to Windows Authentication.

Click Connect.



3. In the SQL Server Management Studio, expand Databases, and click "New Query" (New Query is used for all interactive commands, not just SQL Query statements).

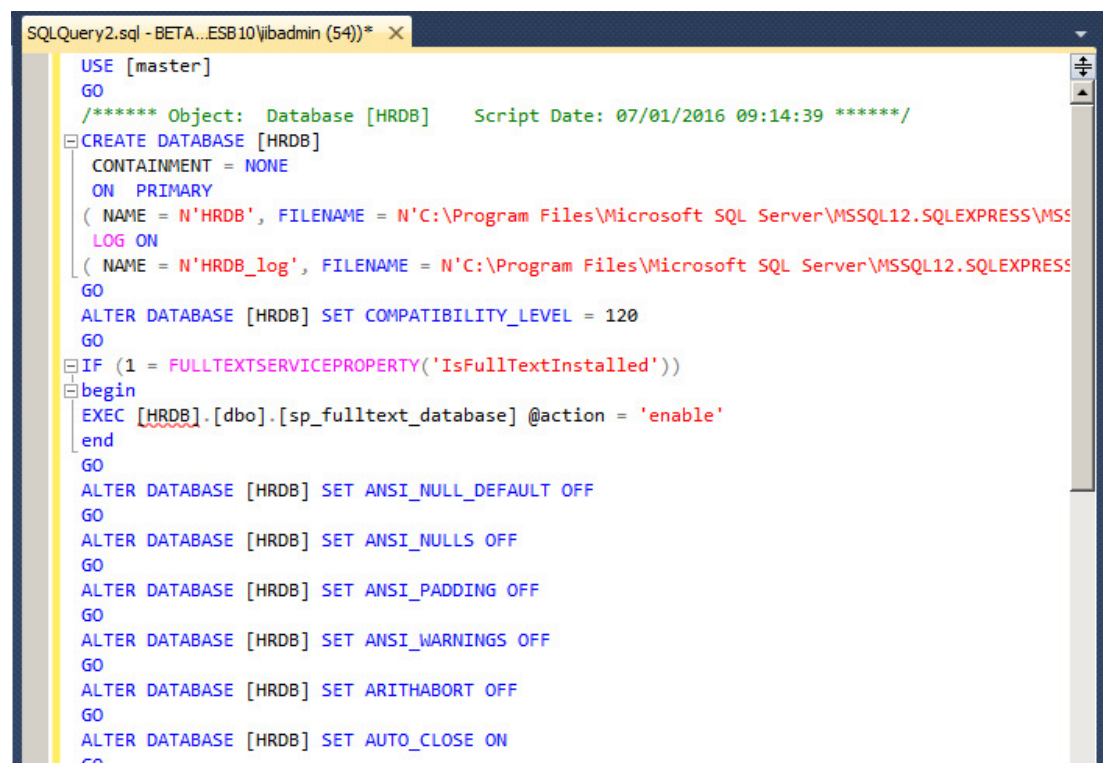


4. Copy / paste the entire contents of the file

`C:\student10\integration_service\SQLServer\DBSetup\HRDB.sql`

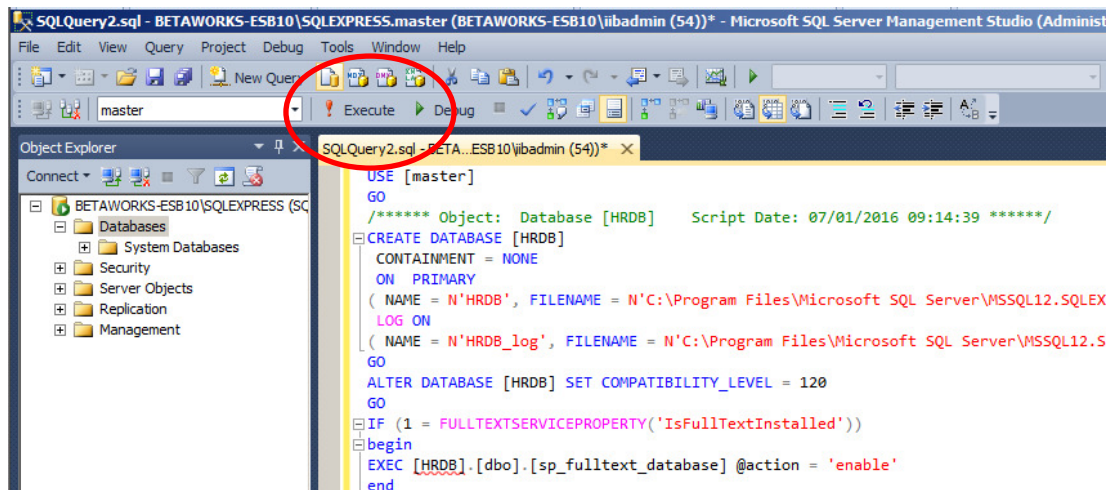
into the Query pane.

This script creates the HRDB database, and the EMPLOYEE table, and inserts 42 rows into the EMPLOYEE table.

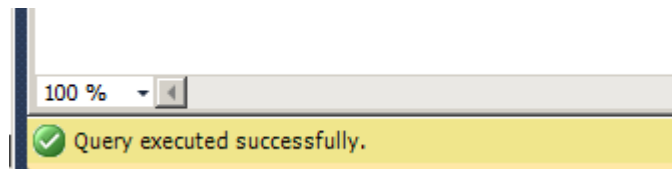




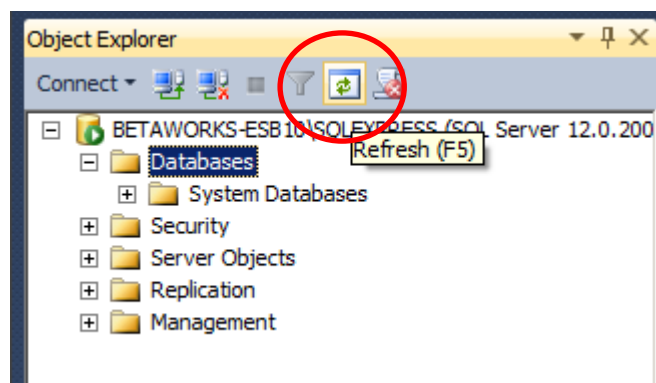
5. Click Execute.



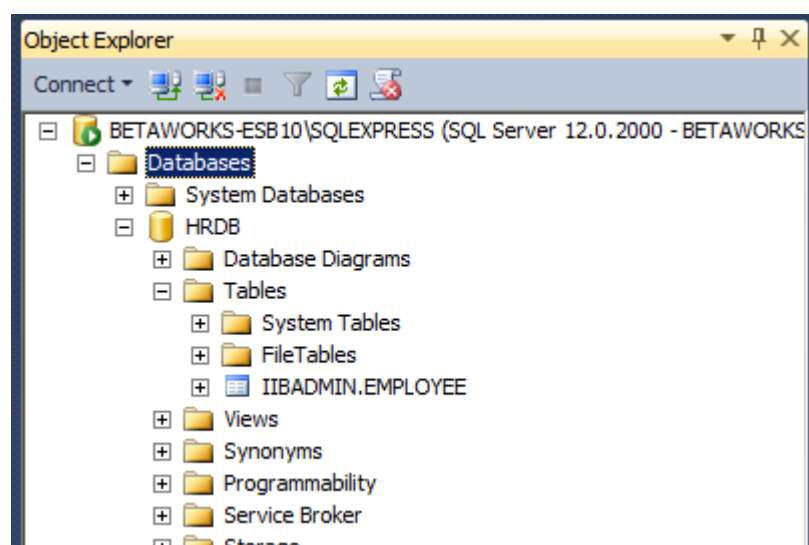
6. The database create and insert statements will be executed, hopefully successfully.



7. Click Databases, and click the Refresh button.



The HRDB database will be shown in the list of databases.



8. Clear the Query pane.

In the Query pane, type

```
Select * from IIBADMIN.EMPLOYEE;
```

Click Execute.

The rows from the EMPLOYEE table should be returned as shown below.

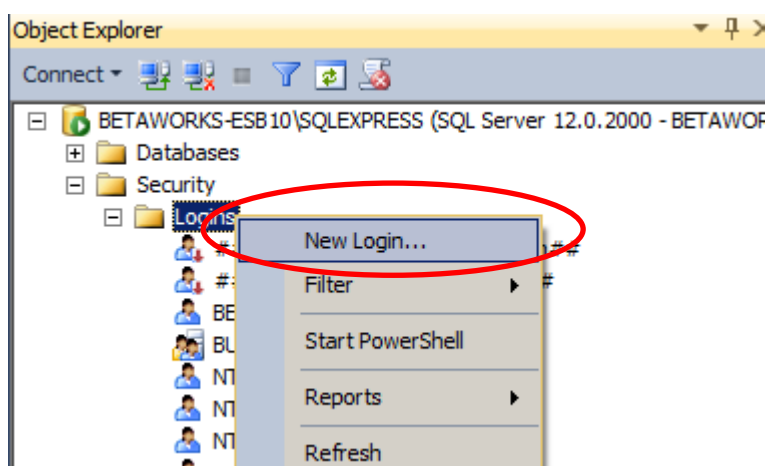
|    | EMPNO  | FIRSTNAME | MIDINIT | LASTNAME  | WORKDEPT | PHONENO | HIREDATE   | JOB      | EDLEVEL | SEX |
|----|--------|-----------|---------|-----------|----------|---------|------------|----------|---------|-----|
| 1  | 000010 | CHRISTINE | I       | HAAS      | A00      | 3978    | 1995-01-01 | PRES     | 18      | F   |
| 2  | 000020 | MICHAEL   | L       | THOMPSON  | B01      | 3476    | 2003-10-10 | MANAGER  | 18      | M   |
| 3  | 000030 | SALLY     | A       | KWAN      | C01      | 4738    | 2005-04-05 | MANAGER  | 20      | F   |
| 4  | 000050 | JOHN      | B       | GEYER     | E01      | 6789    | 1979-08-17 | MANAGER  | 16      | M   |
| 5  | 000060 | IRVING    | F       | STERN     | D11      | 6423    | 2003-09-14 | MANAGER  | 16      | M   |
| 6  | 000070 | EVA       | D       | PULASKI   | D21      | 7831    | 2005-09-30 | MANAGER  | 16      | F   |
| 7  | 000090 | EILEEN    | W       | HENDERSON | E11      | 5498    | 2000-08-15 | MANAGER  | 16      | F   |
| 8  | 000100 | THEODORE  | Q       | SPENSER   | E21      | 0972    | 2000-06-19 | MANAGER  | 14      | M   |
| 9  | 000110 | VINCENZO  | G       | LUCCHESSI | A00      | 3490    | 1988-05-16 | SALESREP | 19      | M   |
| 10 | 000120 | SEAN      |         | CONNELL   | A00      | 2167    | 1993-12-05 | CLERK    | 14      | M   |
| 11 | 000130 | DELORES   | M       | QUINTANA  | C01      | 4578    | 2001-07-28 | ANALYST  | 16      | F   |
| 12 | 000140 | HEATHER   | A       | NICHOLLS  | C01      | 1793    | 2006-12-15 | ANALYST  | 18      | F   |
| 13 | 000150 | BRUCE     |         | ADAMSON   | D11      | 4510    | 2002-02-12 | DESIGNER | 16      | M   |
| 14 | 000160 | ELIZABETH | R       | PIANKA    | D11      | 3782    | 2006-10-11 | DESIGNER | 17      | F   |
| 15 | 000170 | MASATOSHI | J       | YOSHIMURA | D11      | 2890    | 1999-09-15 | DESIGNER | 16      | M   |
| 16 | 000180 | MARILYN   | S       | SCOUTTEN  | D11      | 1682    | 2003-07-07 | DESIGNER | 17      | F   |
| 17 | 000190 | JAMES     | H       | WALKER    | D11      | 2986    | 2004-07-26 | DESIGNER | 16      | M   |
| 18 | 000200 | DAVID     |         | BROWN     | D11      | 4501    | 2002-03-03 | DESIGNER | 16      | M   |

Query executed successfully. | BETAWORKS-ESB10\SQLEXPRESS ... | BETAWORKS-ESB10\iibadm... | HRDB | 00:00:00 | 42 rows

## 2.3 Configure Security

### 2.3.1 Create login credentials for iibuser

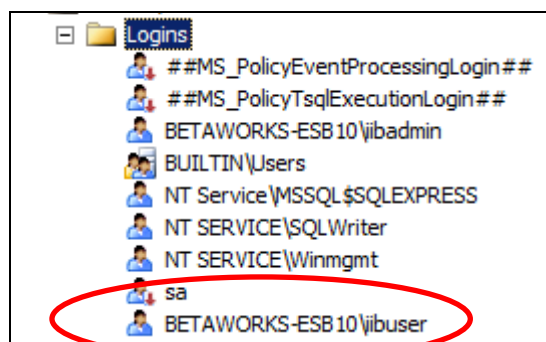
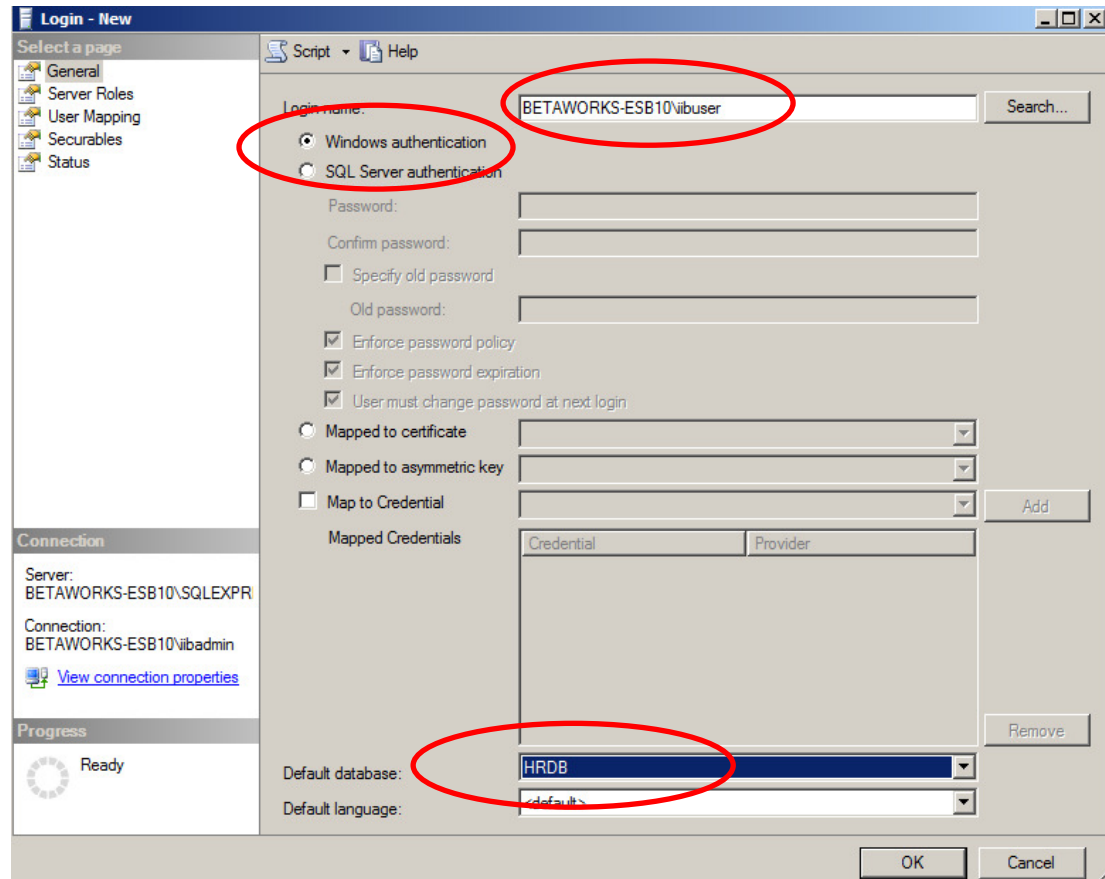
1. Again with the **iibadmin** user, in the Server Management Studio, expand Security, Logins. Right-click Logins, and select New Login.



## 2. Select the General tab, and set the following properties:

- Login name = BETAWORKS-ESB10\ibuser (use the Search button to search for "ibuser" to use the correct format of the name).
- Windows authentication = selected
- Default database = HRDB

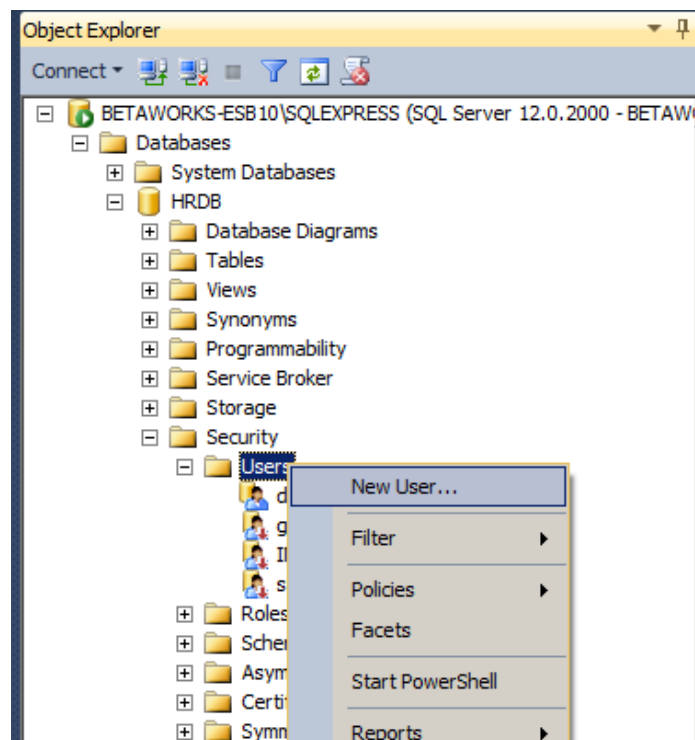
Click OK; the user will be added to the SQL Server Logins.



## 2.3.2 Define database authorisation rules for HRDB

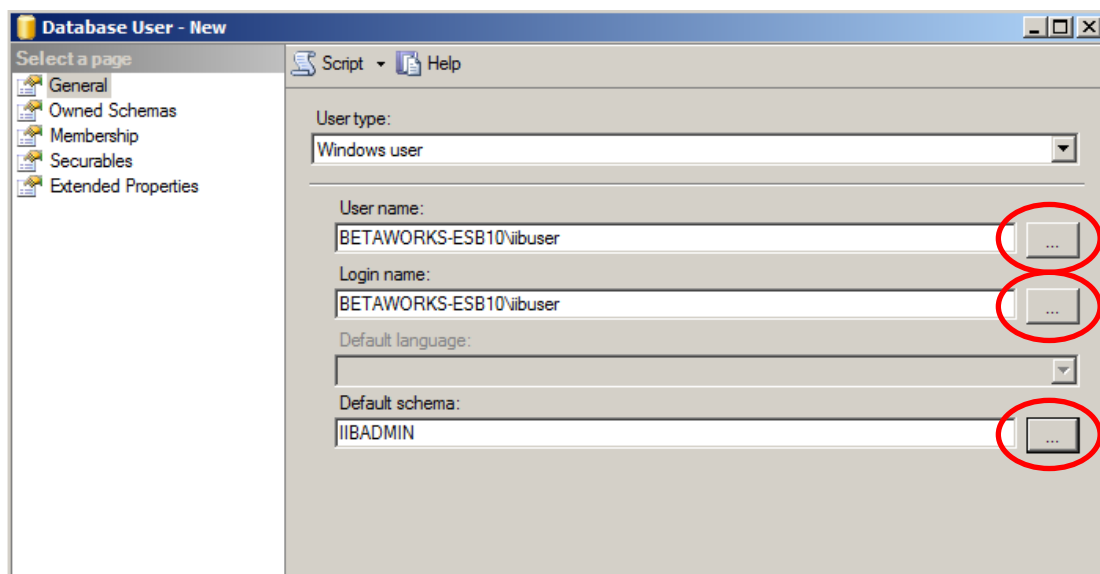
1. Expand Databases, HRDB, Security, Users.

Right-click Users and click "New User".



2. Select the General tab, and set the following properties. Use the "... " button on each property to search for the correct format of the property value.

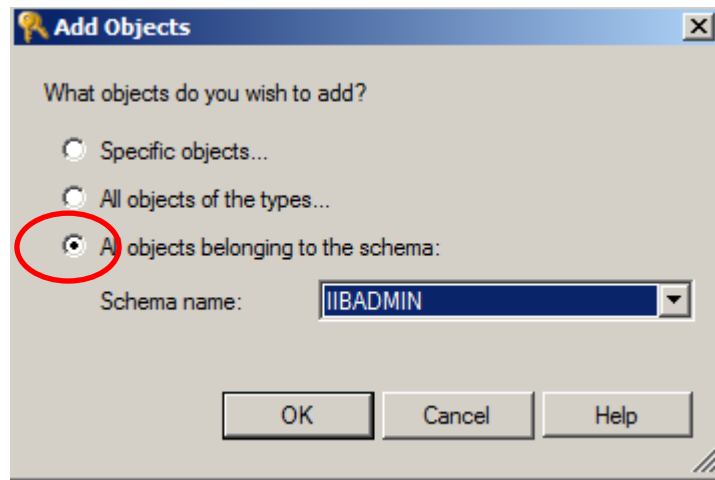
- User type = Windows user
- User name = BETAWORKS-ESB10\iibuser
- Login name = BETAWORKS-ESB10\iibuser
- Default schema = IIBADMIN



3. Select the Securables tab. Click the Search button.

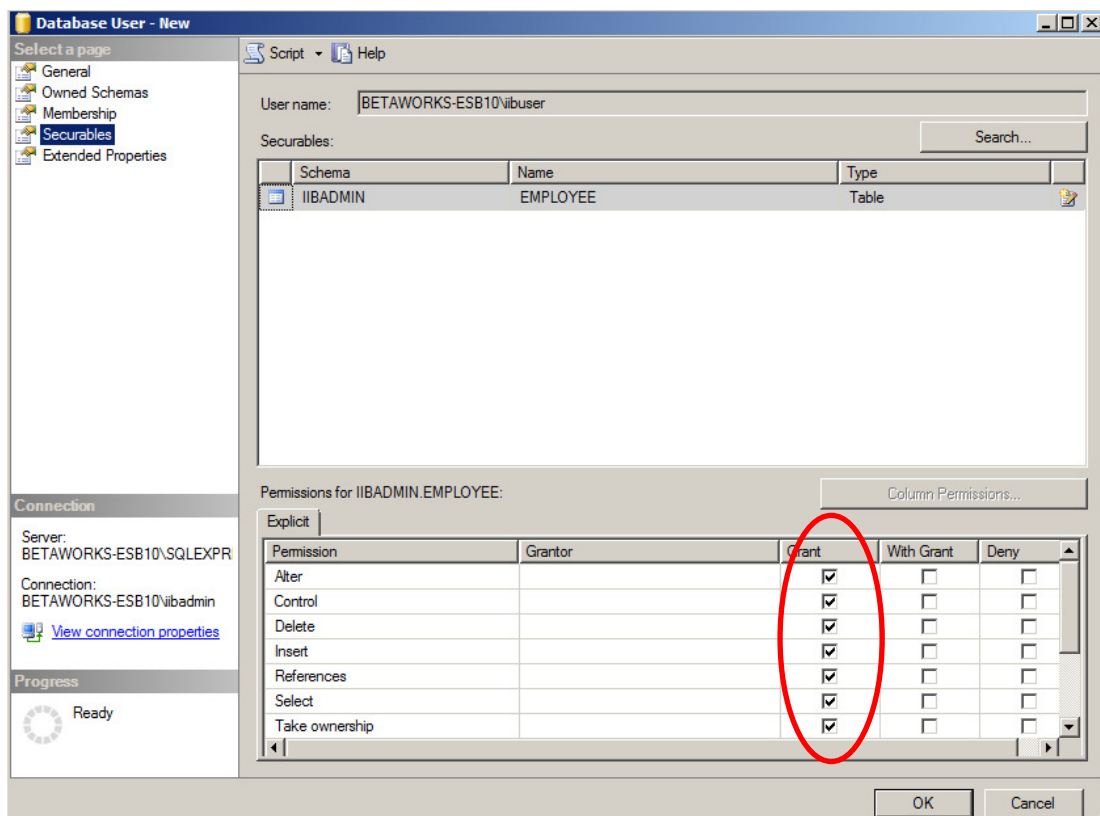
Select "All objects belonging to the schema", and set the Schema name = IIBADMIN.

Click OK.



4. In the Permissions area, tick all the Grant boxes for all types of access.

Click OK to add these database permissions to the BETAWORKS-ESB10\iibuser user.

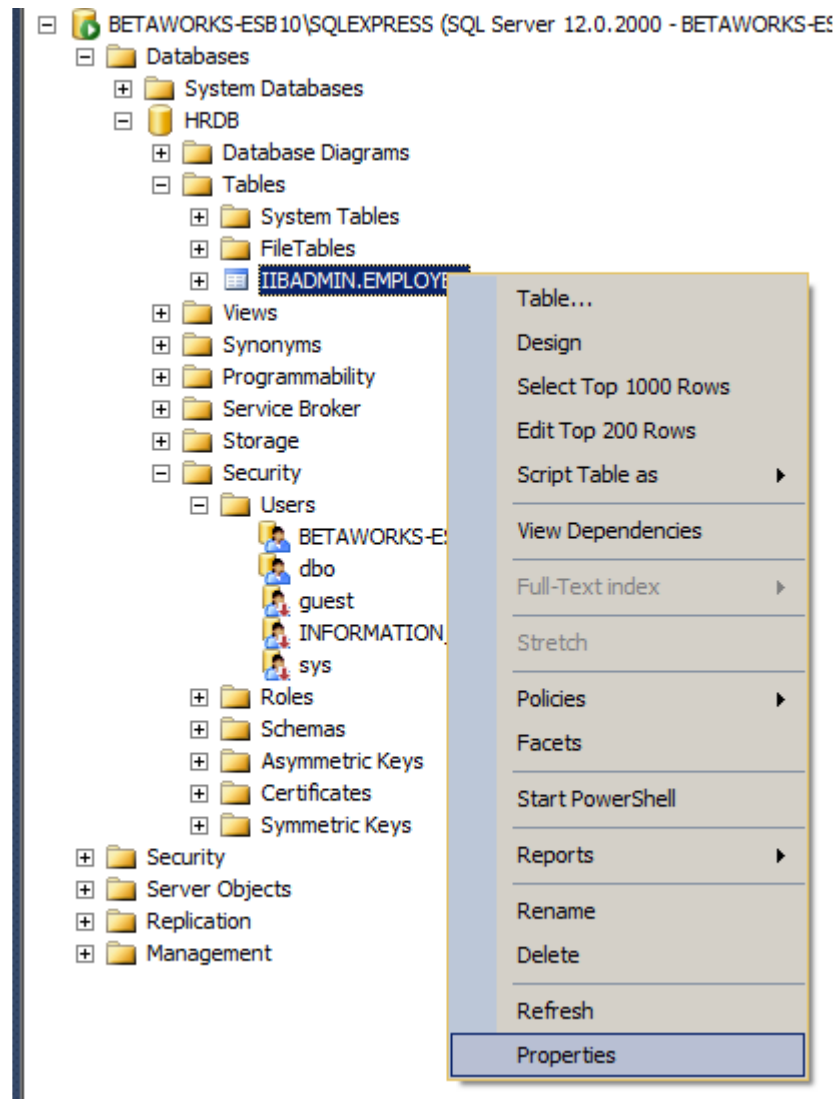


### 2.3.3 Validate database schema and table permissions

In the previous section, you set the required database permissions for the user iibuser. You will now check these permissions using the database views.

1. Expand Databases, HRDB, Tables.

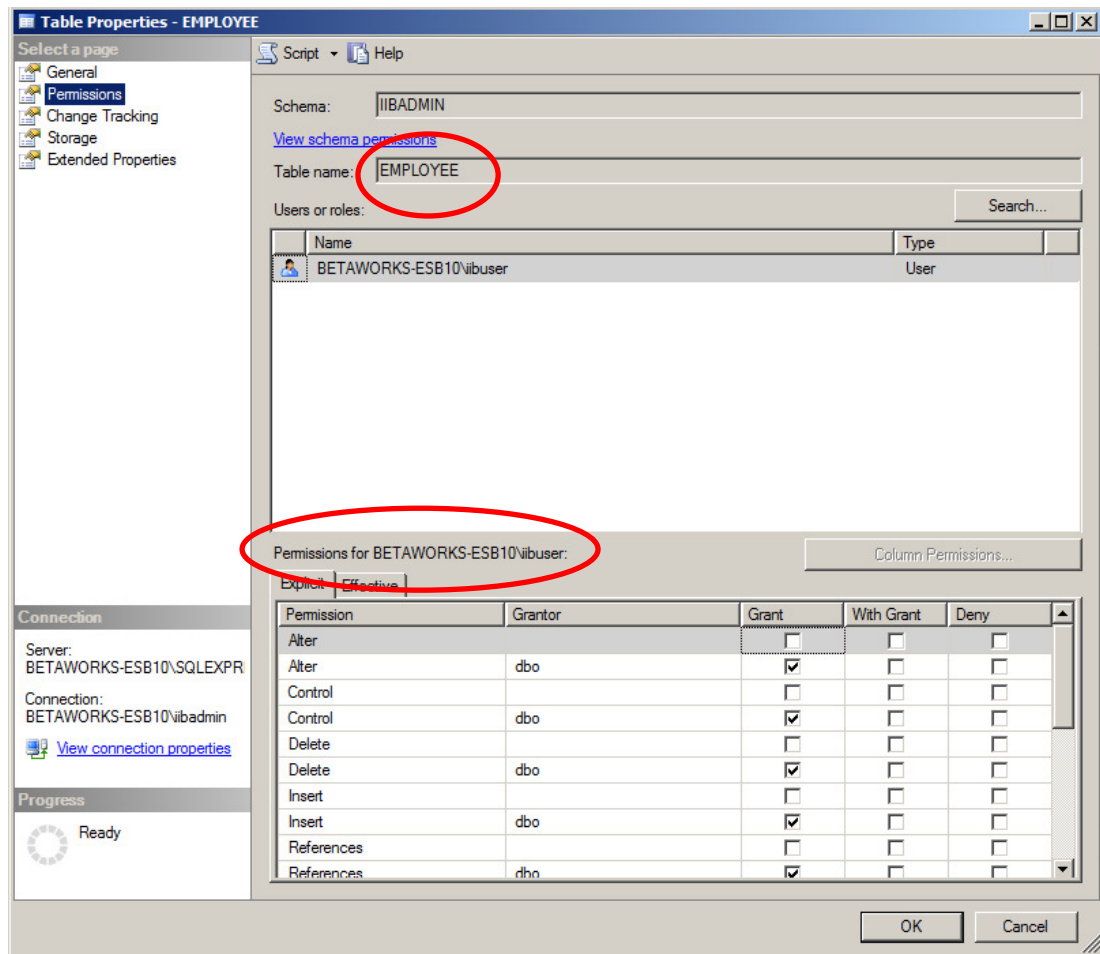
Right-click IIBADMIN.EMPLOYEE, and select Properties.





## 2. Select the Permissions tab.

You will see that the EMPLOYEE table shows BETAWORKS-ESB10\ibuser as a user who has full access rights.

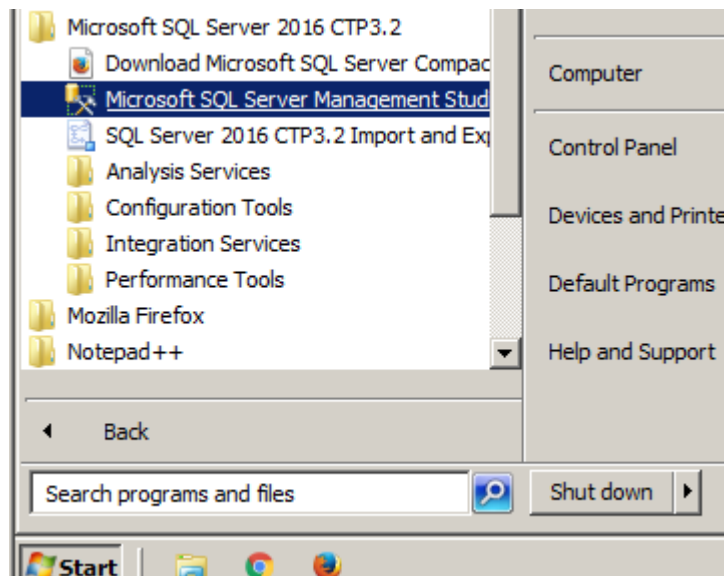


### 2.3.4 Check access from iibuser

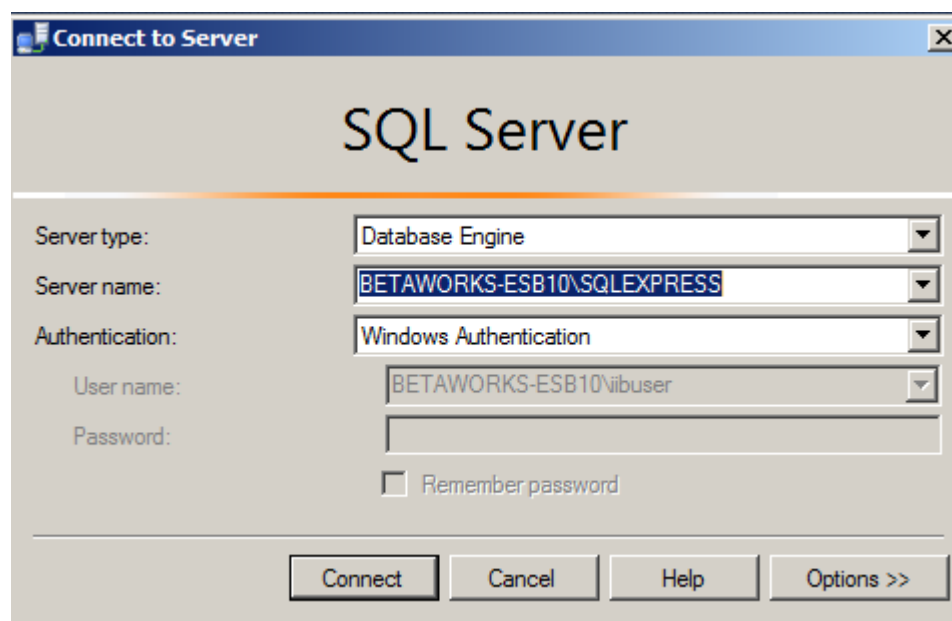
Finally, to validate that the permissions have been specified correctly, use the user **iibuser** to access the HRDB database.

1. Login to Windows with the user **iibuser**, password = **passw0rd**.  
(Use Windows, Start, Switch user).
2. From the Start menu, open Microsoft SQL Server 2016 CTP3.2.

Open "Microsoft SQL Server Management Studio".



3. Click Connect.

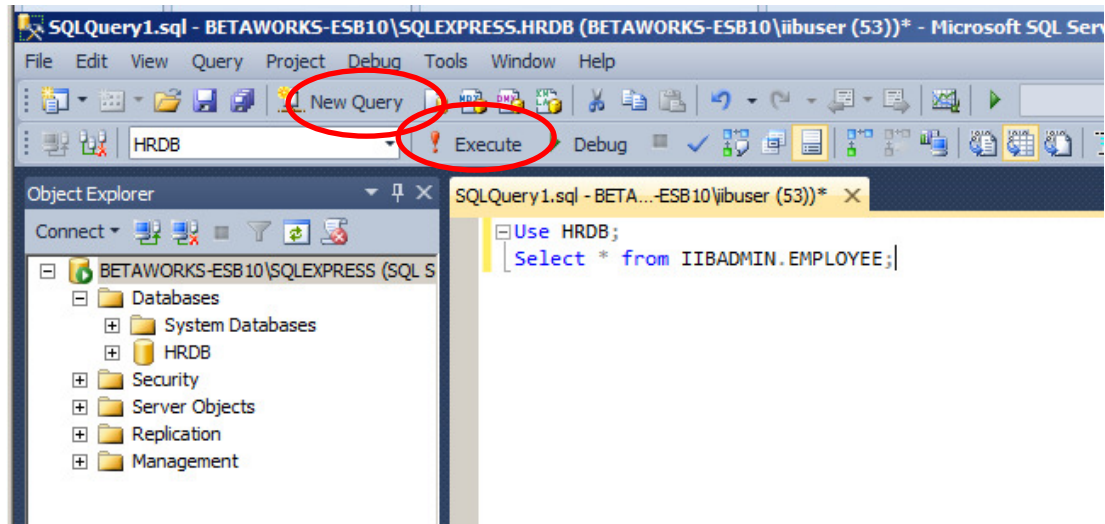


- Click New Query.

In the query pane, enter :

```
Use HRDB;
Select * from IIBADMIN.EMPLOYEE;
```

Click Execute.



- You should see the returned rows from the EMPLOYEE table.

This has confirmed that **iibuser** has the correct permissions to access the EMPLOYEE table in the HRDB database.

|   | EMPNO  | FIRSTNAME | MIDINIT | LASTNAME  | WORKDEPT | PHONENO | HIREDATE   | JOB      | EDLEVEL | SEX |
|---|--------|-----------|---------|-----------|----------|---------|------------|----------|---------|-----|
| 1 | 000010 | CHRISTINE | I       | HAAS      | A00      | 3978    | 1995-01-01 | PRES     | 18      | F   |
| 2 | 000030 | SALLY     | A       | KWAN      | C01      | 4738    | 2005-04-05 | MANAGER  | 20      | F   |
| 3 | 000050 | JOHN      | B       | GEYER     | E01      | 6789    | 1979-08-17 | MANAGER  | 16      | M   |
| 4 | 000060 | IRVING    | F       | STERN     | D11      | 6423    | 2003-09-14 | MANAGER  | 16      | M   |
| 5 | 000070 | EVA       | D       | PULASKI   | D21      | 7831    | 2005-09-30 | MANAGER  | 16      | F   |
| 6 | 000090 | EILEEN    | W       | HENDERSON | E11      | 5498    | 2000-08-15 | MANAGER  | 16      | F   |
| 7 | 000100 | THEODORE  | Q       | SPENSER   | E21      | 0972    | 2000-06-19 | MANAGER  | 14      | M   |
| 8 | 000110 | VINCENZO  | G       | LUCCHESI  | A00      | 3490    | 1988-05-16 | SALESREP | 19      | M   |

- No further use will be made of the Server Management Studio, so close it now.

## 3. Create and configure the IIB node and application

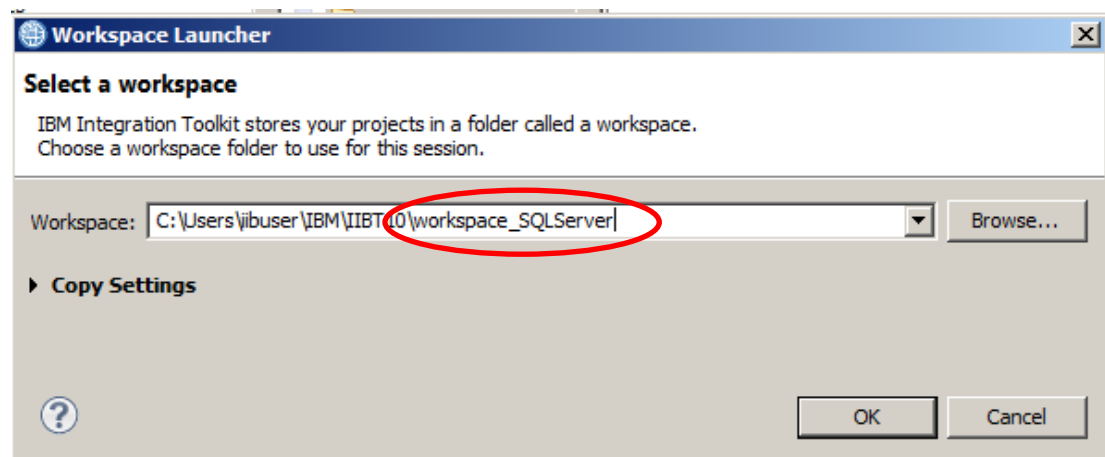
### 3.1 Create IIB node

In previous labs in this series, you have used the generated TESTNODE\_iibuser node to run the IIB applications and services. The TESTNODE node is created automatically by the IIB Toolkit, and uses a SYSTEM user as the owning user (ie. the corresponding "mqsi" command does not include the "-i iibuser" parameter).

However, when using Windows integrated authentication, it is necessary to create a new IIB node which uses a specific user, not the SYSTEM user.

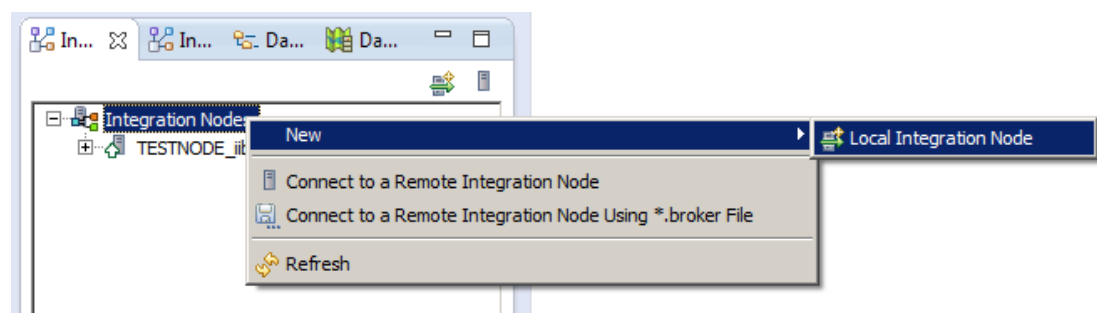
1. Login to Windows with the user **iibuser**, password = **passw0rd**. (You are already there, if you are following the steps in this lab guide).
2. Start the IIB Toolkit, and create a new workspace, for example with the name "workspace\_SQLServer", as shown.

Click OK.



3. Create a new Local Integration Node:

Right-click Integration Nodes, New, Local Integration Node.



## 4. Set the following properties:

- Name = IB10NODE\_SQLS
- User name = iibuser
- Password = passw0rd

Click Finish.

The new node will take a few moments to create. When complete, click Close.

**New Local Integration Node**

Create a new local integration node

**Integration Node**

\*Integration node name: IB10NODE\_SQLS

\*User name: iibuser

\*Password: .....

☒ Start the integration node when Windows starts

**Server**

\*Default integration server name: default

Default debug port:

**Default Queue Manager**

Configure a local queue manager to be used by all the nodes that do not explicitly specify MQ connection

Queue manager name:

Note that if you plan to use any of the following nodes in message flows or subflows, you will need to run the iib\_createqueues script that creates the necessary system queues: AggregateControl, AggregateReply, AggregateRequest, Collector, Resequence, Sequence, TimeoutControl, TimeoutNotification

Finish Cancel

## 5. Create a Configurable Service for the JDBC connection to the HRDB SQL Server database.

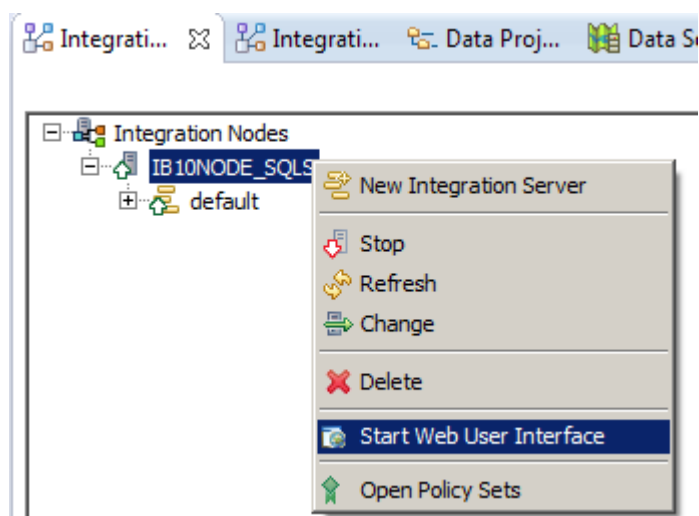
In an IIB Command Console, run the following commands:

```
cd \student10\integration_service_SQLServer\ConfigService
create_JDBC_for_HRDB_SQLSERVER
```

Accept the default node (IB10NODE\_SQLS) for the command.

6. Review the configurable service you just created using a web browser.

In the IIB Toolkit, right-click the node IB10NODE\_SQLS to automatically open a web admin browser connected to this node.



## 7. Expand Operational Policy, Configurable Services, JDBC Providers, and select HRDB.

Review the following properties carefully, which have been changed from the default values:

- databaseType - Microsoft SQL Server
- jdbcProviderXASupport - false (IIB does not support the XA protocol with SQL Server databases. If you leave this value set to true, you will observe JDBC XA errors when the service is executed).
- portNumber - set to the SQL Server TCP/IP port you set earlier.
- jarsURL - this is the directory name that holds the Microsoft JDBC drivers. Note that this is not included in the main SQL Server package, and has to be downloaded and installed (unzipped) separately.
- databaseName - HRDB
- securityIdentity - This property is not used in this lab scenario, because we have chosen to use Windows integrated security (see next property). If we had configured the lab in the same way as the DB2 version, we would normally set this property to a Security Identity defined by a corresponding mqsisetdbparms command.
- connectionUrlFormat - This property has been changed to replace the string "user=[user],password=[password]", with the string "**integratedSecurity=true**". This means that the user credentials that are used to start the IIB node are used to authenticate and authorise access to the SQL Server system automatically by Windows. (The user is iibuser in this lab).

**HRDB - JDBCProviders Configurable Service**

Overview Edit

▼ Properties

|                          |  |
|--------------------------|--|
| type4DatasourceClassName | com.microsoft.sqlserver.jdbc.SQLServerXADataSource   |
| type4DriverClassName     | com.microsoft.sqlserver.jdbc.SQLServerDriver   |
| databaseType             | Microsoft SQL Server   |
| jdbcProviderXASupport    | false  |
| portNumber               | 49688  |
| connectionUrlFormatAttr5 |  |
| connectionUrlFormatAttr4 |  |
| serverName               | localhost  |
| connectionUrlFormatAttr3 |  |
| connectionUrlFormatAttr2 |  |
| connectionUrlFormatAttr1 |  |
| environmentParms         | default_none   |
| maxConnectionPoolSize    | 0  |
| description              | default_Description  |
| jarsURL                  | C:\Program Files\Microsoft JDBC Driver 4.2 for SQL Server\sqljdbc_4.2\enu  |
| databaseName             | HRDB   |
| databaseVersion          | default_Database_Version   |
| securityIdentity         | None required because we are using Windows Integrated Authentication (integratedSecurity=true instead of username-password security in connectionURL format parameter) |
| connectionUrlFormat      | jdbc:sqlserver://[serverName]:[portNumber];DatabaseName=[databaseName];integratedSecurity=true   |
| databaseSchemaNames      | useProvidedSchemaNames   |

## 3.2 Set Windows Path for SQL Server authorisation for IIB

Both the IIB Toolkit and the IIB runtime node require access to the file **sqljdbc\_auth.dll**. This file is required when using Windows integrated authentication to connect IIB components to the SQL Server database system, which will be the method used in this lab. (Note - this requirement is not restricted to IIB - see multiple items on many SQL Server forums for more discussion).

This file is shipped in the SQL Server JDBC package. In the IIB workshop lab VMWare image, the Windows Path has already been changed to include the directory:

```
c:\Program Files\Microsoft JDBC Driver 4.2 for SQL Server\sqljdbc_4.2\enu\auth\x64
```

You can check this by running the DOS command "Set Path".

If you are running this lab on your own Windows system, you will need to make a similar change, and rebooting your system.



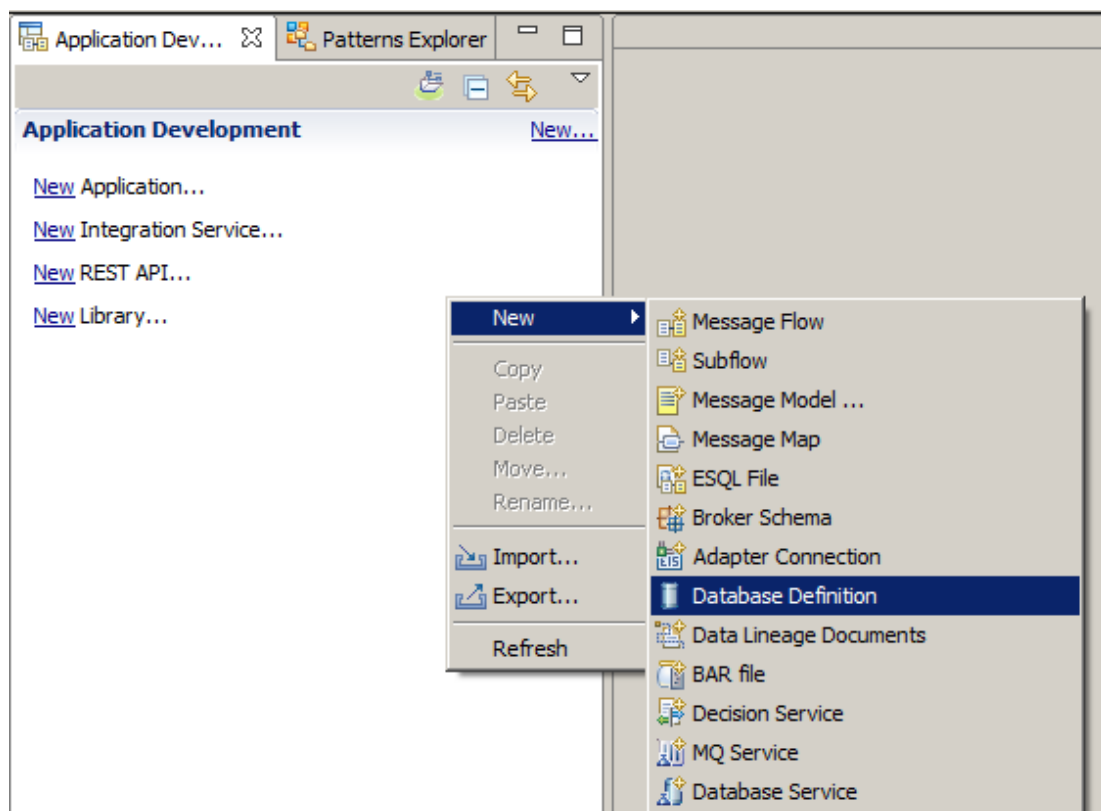
### 3.3 Create Database Definition file for HRDB

In this part of the lab, you will use the IIB Toolkit to connect to the HRDB database on SQL Server, and use this to create an IIB database definition. This would then typically be used to create an IIB Message Model (see Lab 1 in this series).

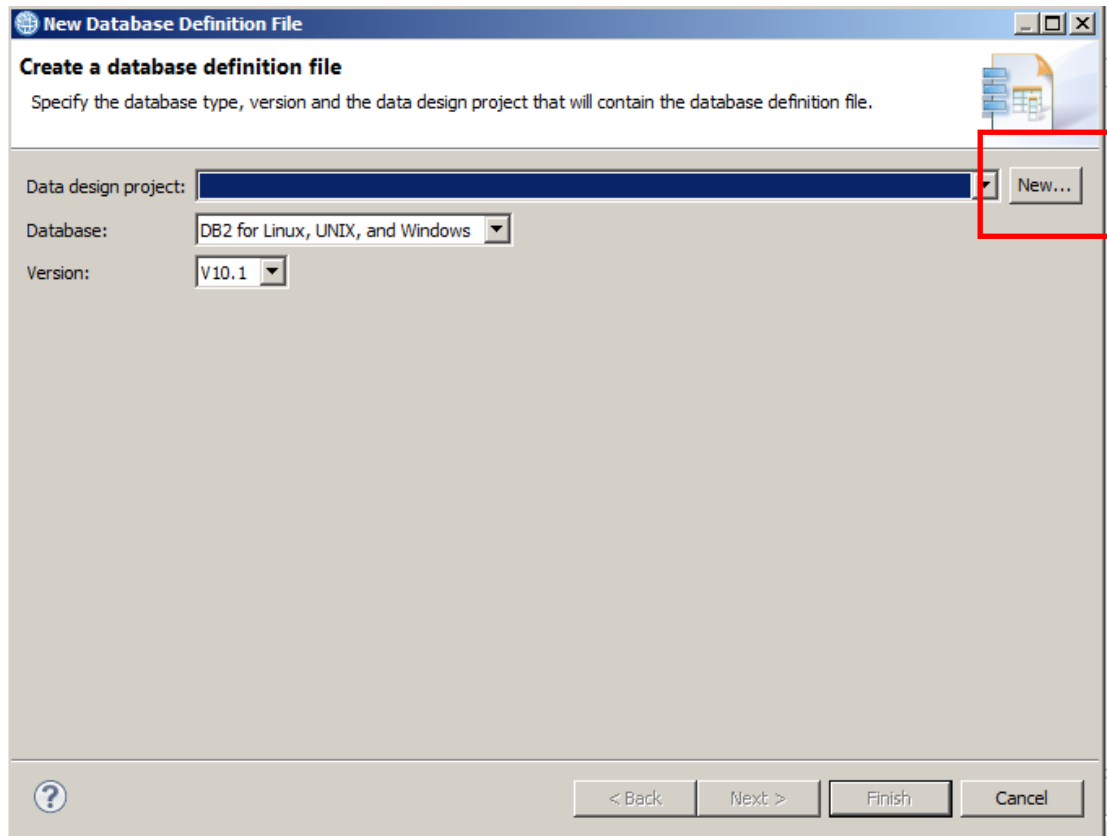
We have included these IIB Toolkit tasks for SQL Server here, because they are a little different from the equivalent tasks when connecting to DB2. However, because the existing EmployeeService used in the DB2 version of this lab has been written, this will be used to perform the runtime part of this lab. Hence, later in this lab, you will import the pre-built solution of EmployeeService (demonstrating portability of IIB applications across different database vendors - the original application was built for use with DB2). The database definition that you are about to create will not actually be used when this service is implemented and tested.

Make sure you are logged in as "**iibuser**", and open the IIB Toolkit using the shortcut on the Windows Start menu (if already open from the previous step, you can use this).

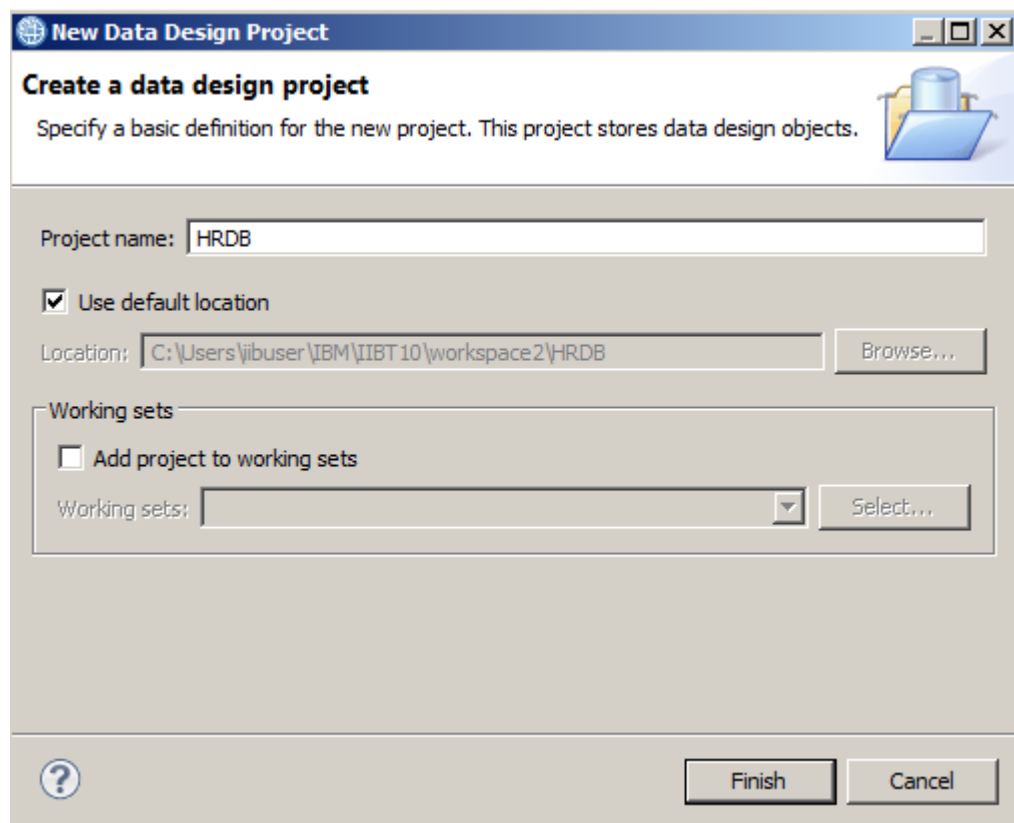
1. In the Toolkit navigator, right-click in blank space, select New, Database Definition.



2. Click New to create a data design project.

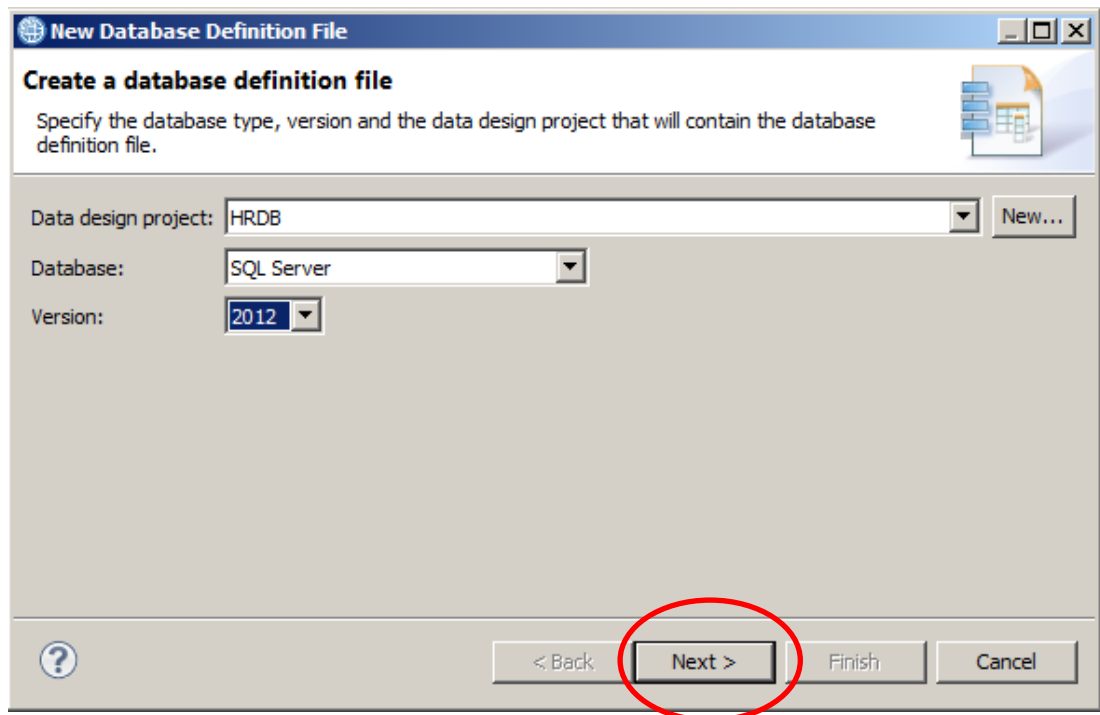


3. Name it HRDB. Click Finish.

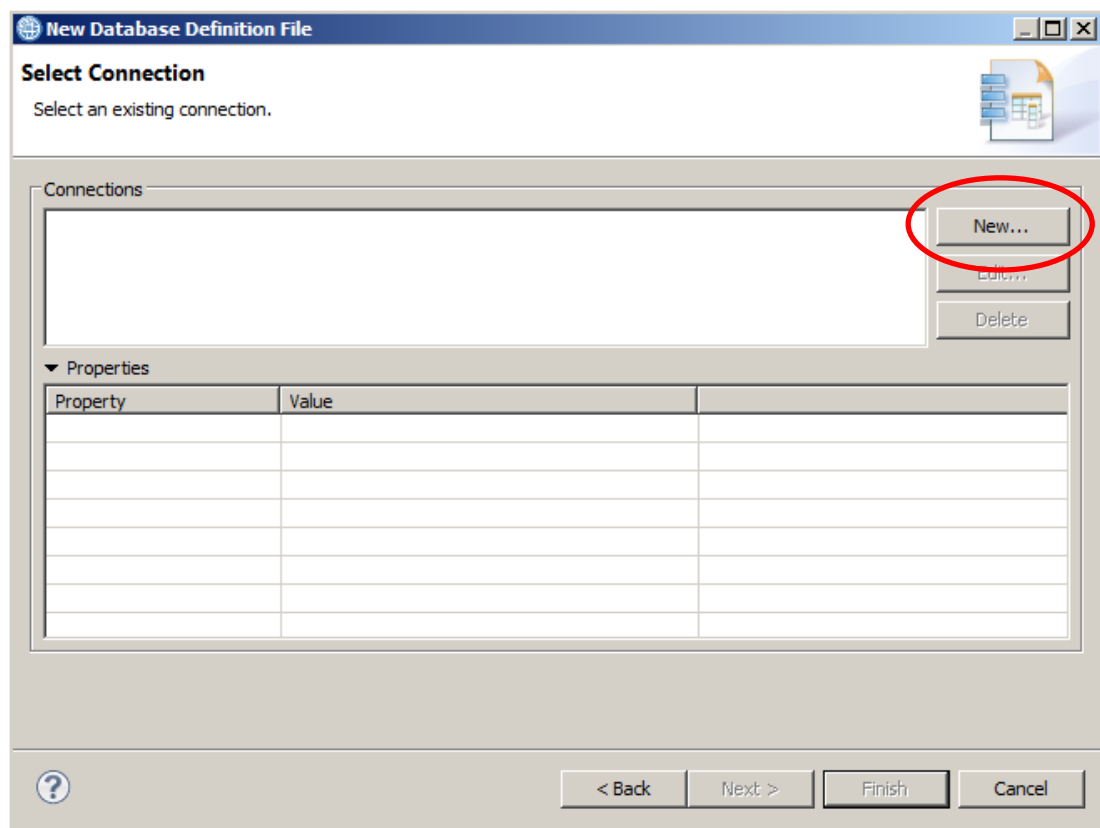


- Set the Database to SQL Server, and the Version to 2012 (2014 is supported and SQL Server Express Edition 2014 is installed on the workshop VM system).

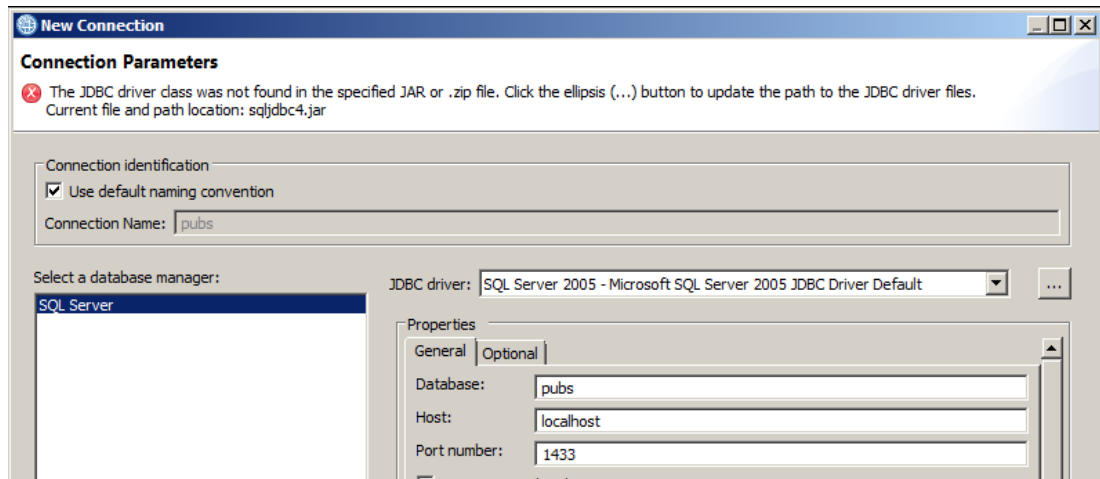
Click Next.



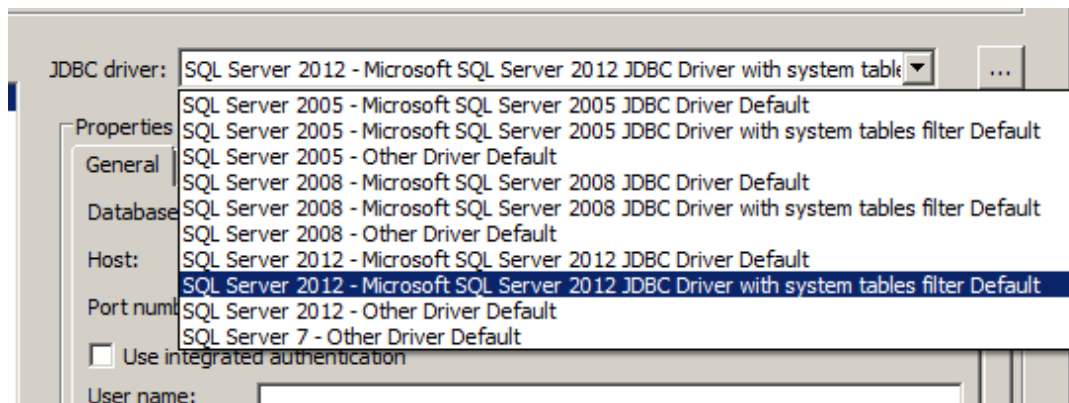
- Click New to create a new connection.



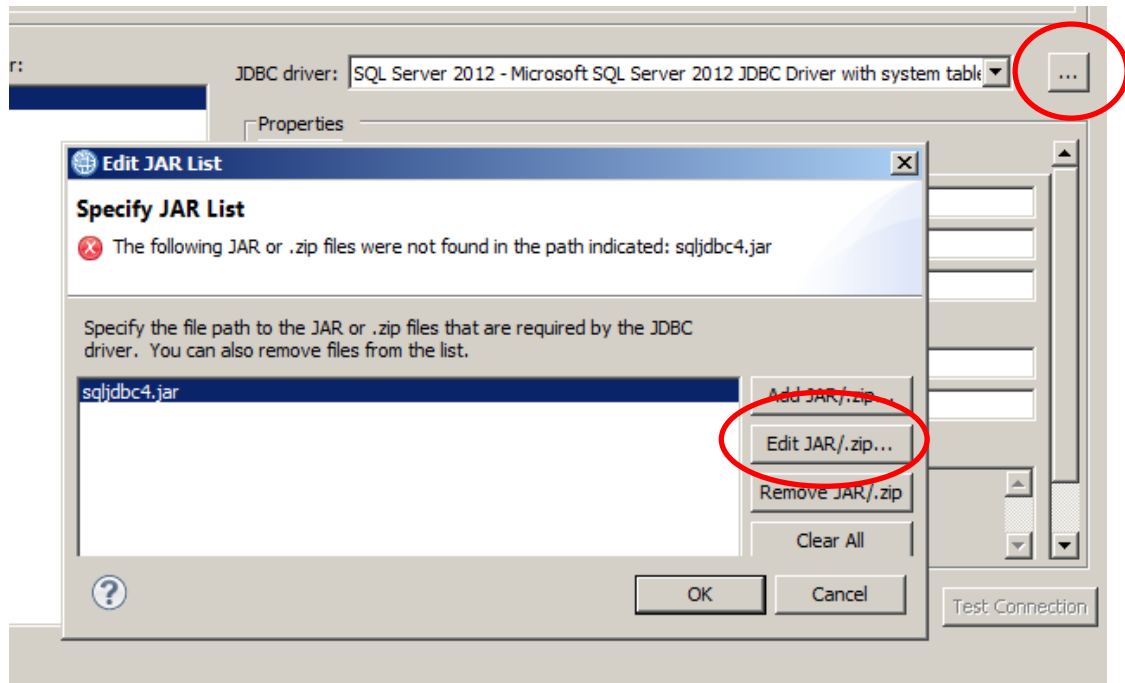
- At the top of the window, you will see a message that says the required JDBC driver class, sqjjdbc4.jar, was not found.



- Click the JDBC driver dropdown, and select "SQL Server 2012 .... with system tables".



8. Click on the three dots button to the right of the JDBC driver dropdown. This will show you the name of the JDBC driver file that the Toolkit is looking for.

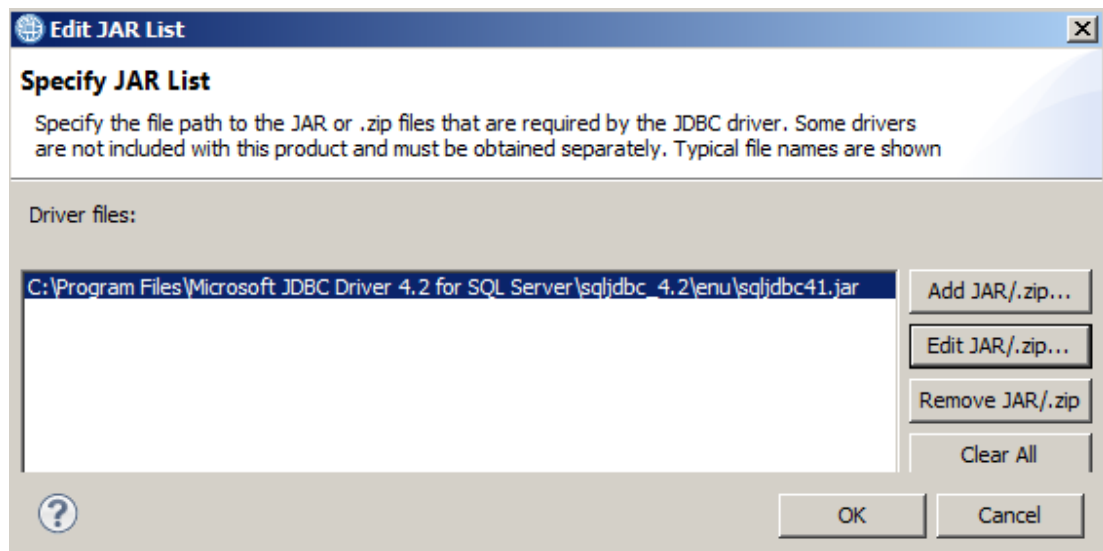


9. Using the "Edit JAR/.zip" button, specify the location of the sqljdbc41.jar file.

On the IIB workshop system, this is

`C:\Program Files\Microsoft JDBC Driver 4.2 for SQL  
Server\sqljdbc_4.2\enu\Sqljdbc41.jar`

Click OK.



10. Set the following properties:

- Database name = HRDB
- Host = localhost
- Port number = 49688 (this must match the change you made earlier for the SQL Server TCPIP port)
- Use integrated authentication = ticked

Click Test Connection, to ensure you have set the right properties.

Assuming the Test Connection is successful, click Finish.

JDBC driver: SQL Server 2012 - Microsoft SQL Server 2012 JDBC Driver with system table

Properties

General Optional

Database: HRDB

Host: localhost

Port number: 49688

☒ Use integrated authentication

User name:

Password:

☐ Save password

Connection URL: jdbc:sqlserver://localhost:49688;databaseName=HRDB;integratedSecurity=true;

Test Connection

11. Highlight the new connection (HRDB1), and note the various connection properties that are shown (confirming the definitions you just made).

Click Next.

New Database Definition File

Select Connection

Select an existing connection.

Connections

HRDB1

New... Edit... Delete

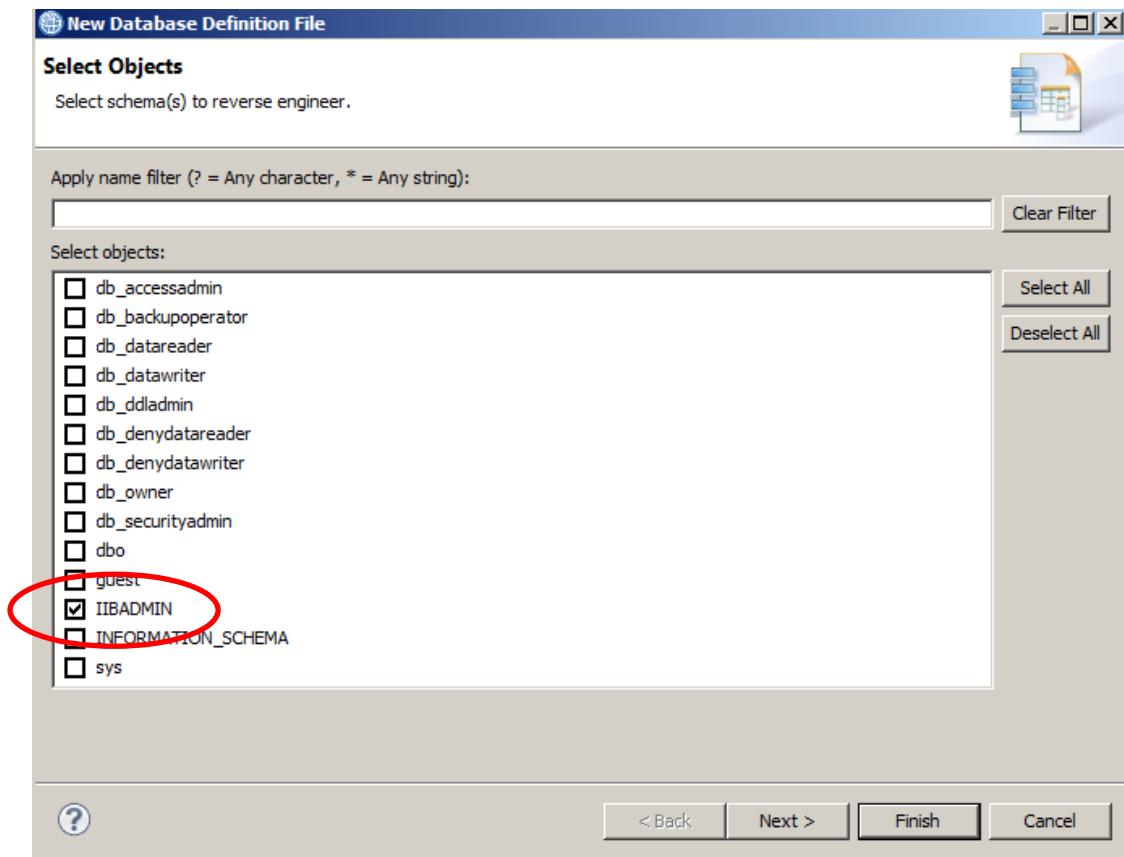
Properties

| Property          | Value   |
|-------------------|---|
| Name              | HRDB1   |
| Description       |   |
| Category          | Database Connections  |
| Database          | HRDB  |
| JDBC Driver Class | com.microsoft.sqlserver.jdbc.SQLServerDriver  |
| Class Location    | C:\Program Files\Microsoft JDBC Driver 4.2 for SQL Server\sqljdbc_4.2\enu\sqljdbc41.jar |
| Connection URL    | jdbc:sqlserver://localhost:49688;databaseName=HRDB;integratedSecurity=true;             |
| User ID           |   |

< Back Next > Finish Cancel

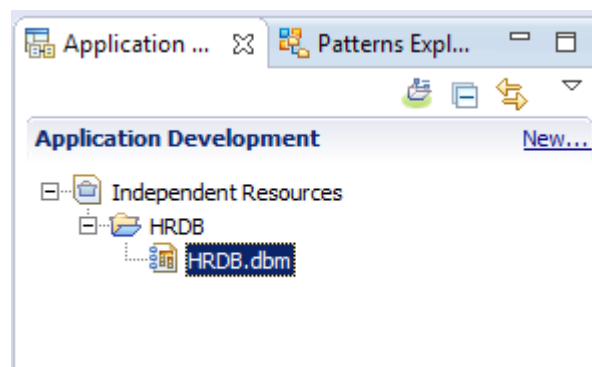
12. Select the IIBADMIN schema, and click Finish.

(Note that the HRDB database has been defined with the IIBADMIN schema. Recall that you earlier granted permissions for this user on this database/schema).



13. 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.



## 3.4 Import the prebuilt integration service

Although we have shown you how to use the IIB Toolkit to connect to a SQL Server database, and create a database definition, the rest of this lab will use the existing integration service that was built in Lab 1 (Employee Service). This application will be used unchanged, and will therefore demonstrate that IIB applications are independent of the connected database, providing that the database properties such as database name, schema name and table structure are the same across the different databases.

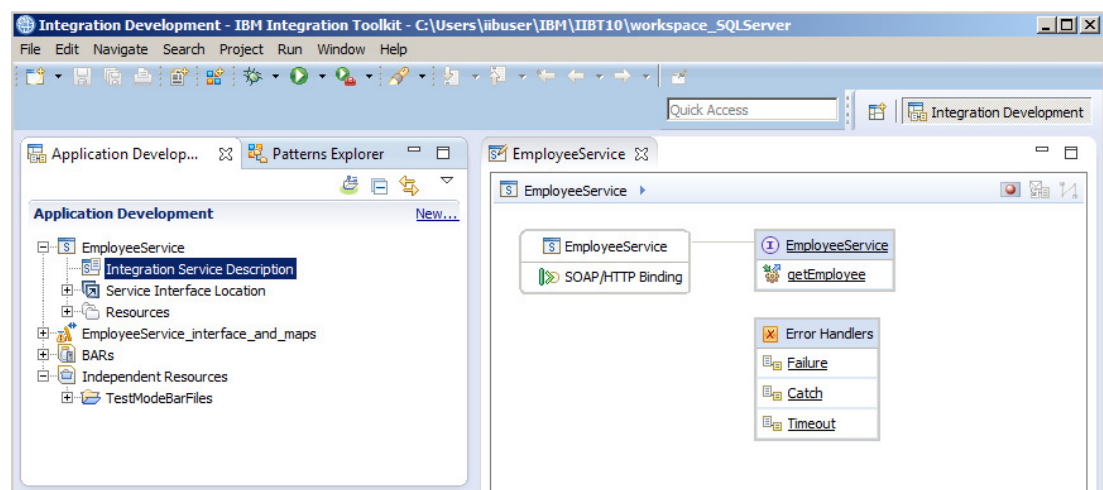
1. In the IIB Toolkit, import the PI file

```
c:\student10\integration_service\solution\  
EmployeeService.10.0.0.3.zip
```

Import all projects. Because you have earlier created a project named HRDB, this will be overwritten with the project HRDB contained in the PI file. The Toolkit will ask you to confirm the over-write of the HRDB project. Answer Yes to this.

The imported projects will be shown like this.

Expand the EmployeeService, and double-click the Integration Service description. This will open the service definition for EmployeeService.



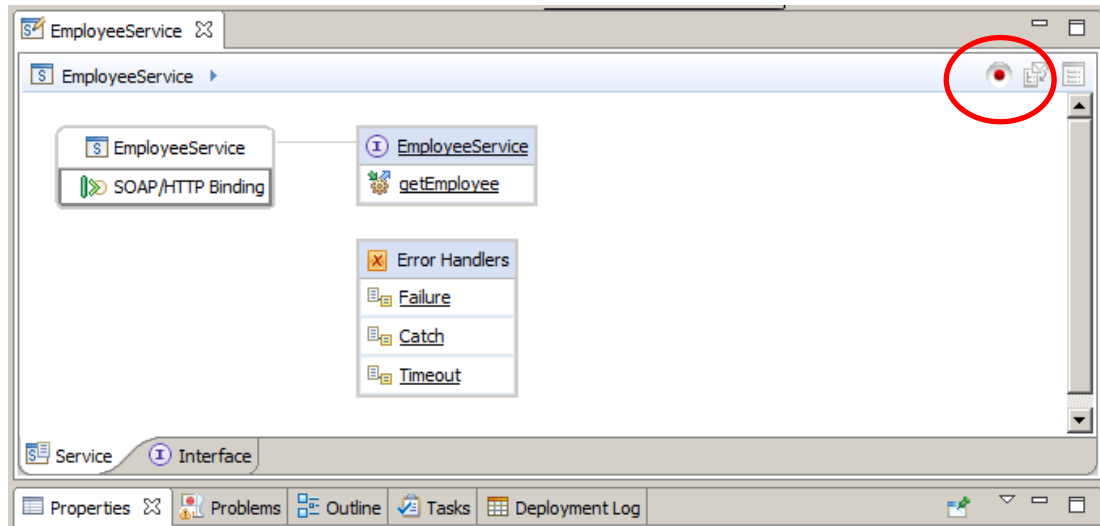


## 4. Test the Integration Service with the Flow Exerciser

This lab will use the Flow Exercise to perform a simple unit test of the service you have just imported.

First, make sure that the node IB10NODE\_SQLS 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.



2. Click the red button. The integration service will be deployed to the IB10NODE\_SQLS/default server.

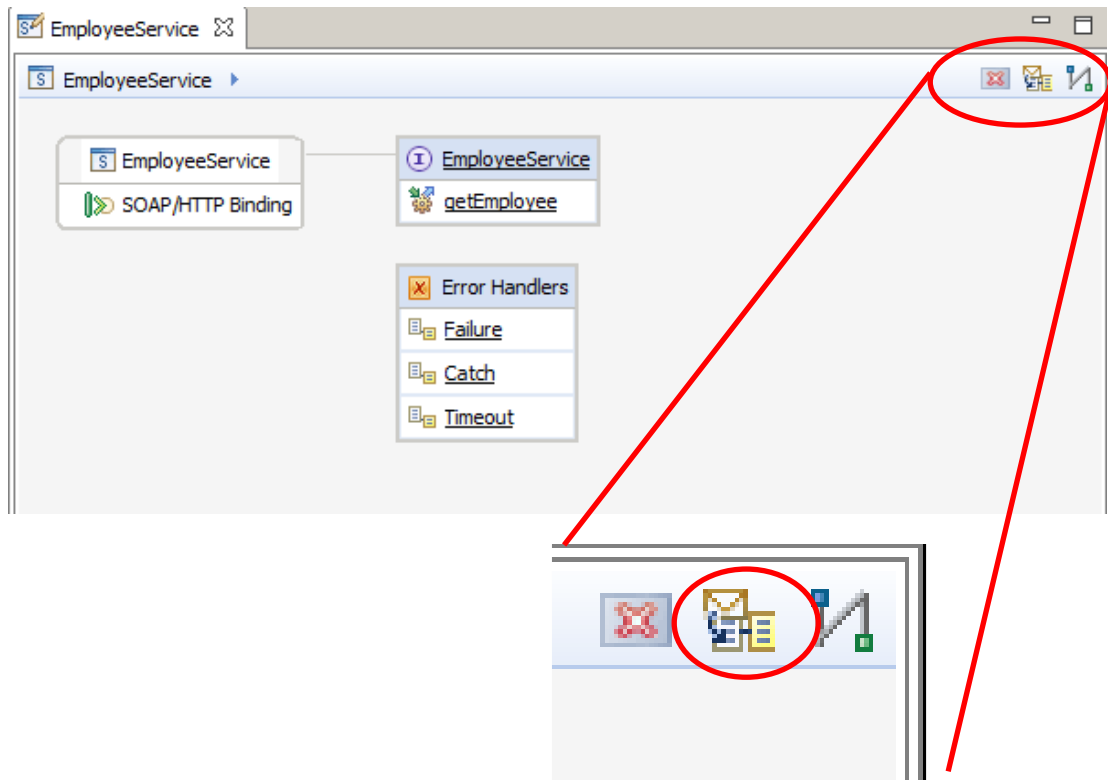
Note that if you have other nodes or servers running, you will be presented with a selection dialogue. Make sure you select the correct node/server.

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.

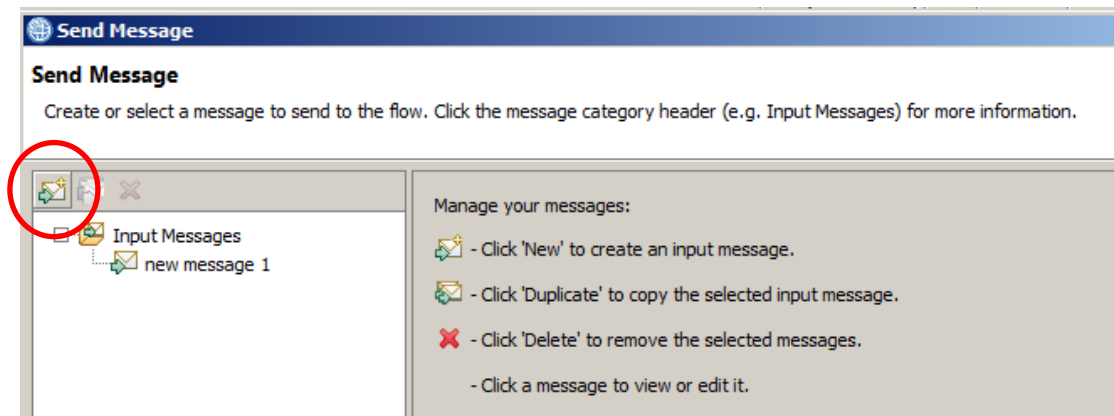


3. The flow editor will turn grey, indicating that the Flow Exerciser is ready for use.

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



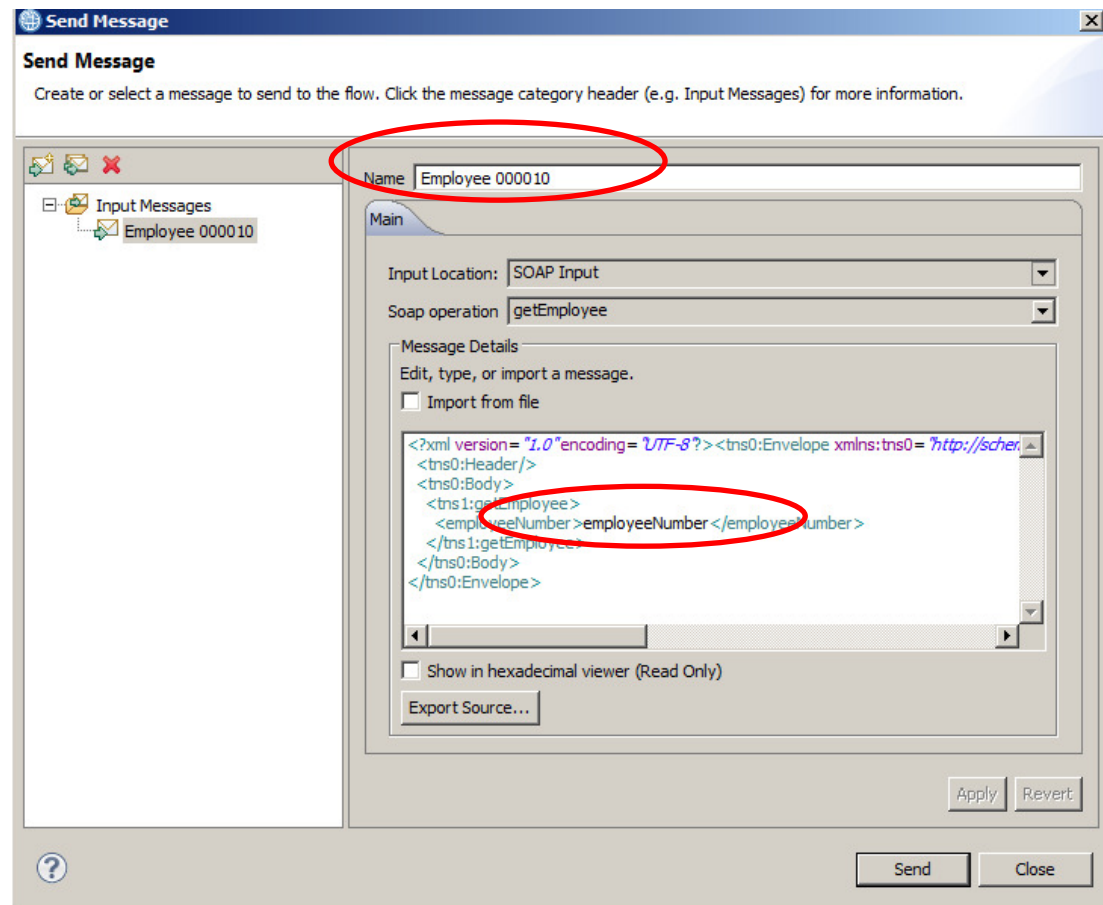
4. In the Send Message window, click the button to create a new message.



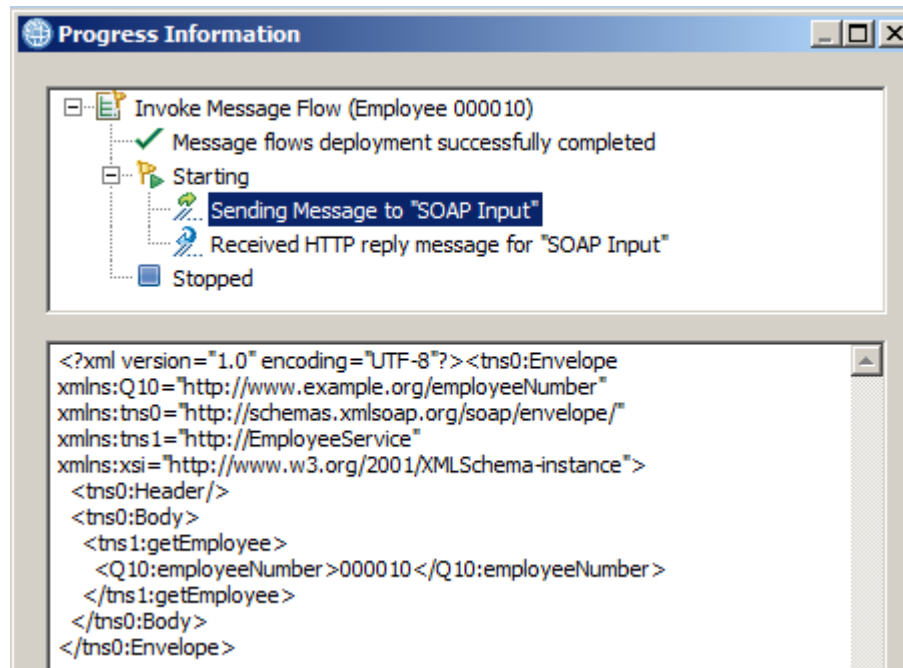
5. Name the new message "Employee 000010".

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.

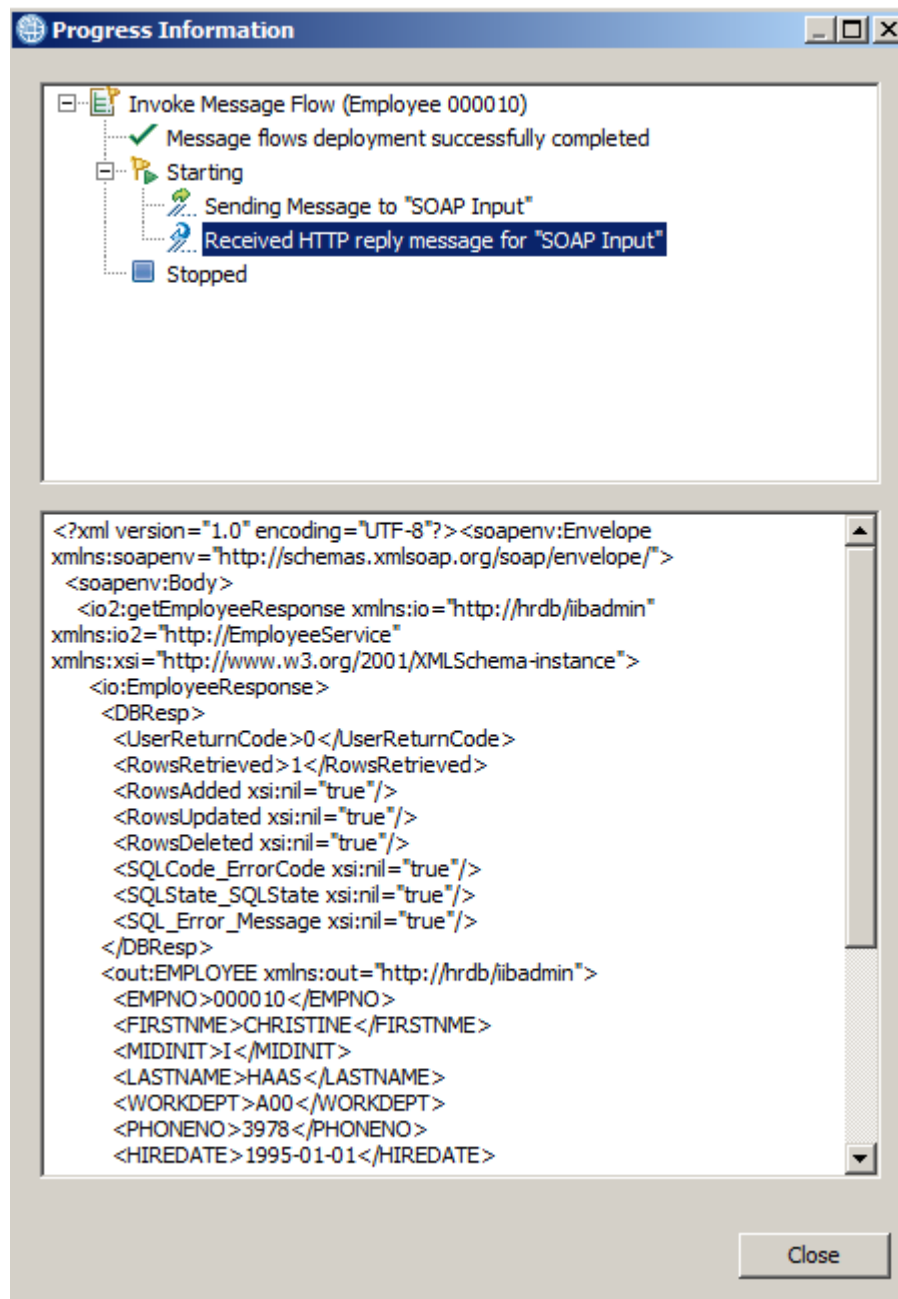


6. The service operation will be executed. Highlighting the line "Sending Message to SOAP Input". This will show the message that was sent to the service.



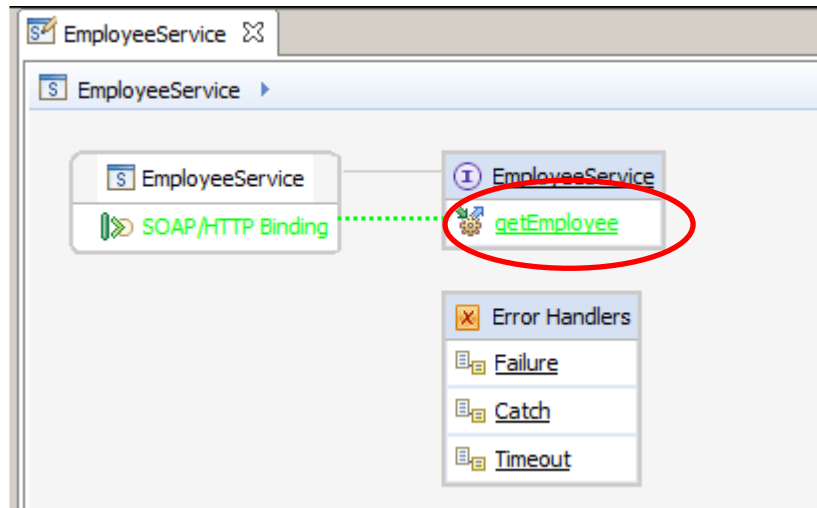
7. Highlighting the "Received HTTP" line will show the message returned from the service. Note that the employee record for the selected employee number has been retrieved from the EMPLOYEE table.

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

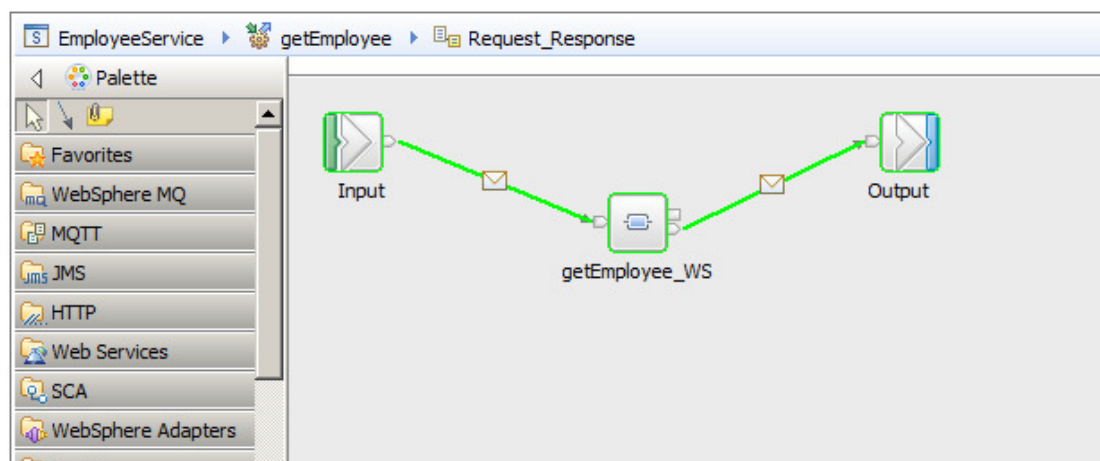


Close the Progress Information window.

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



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.

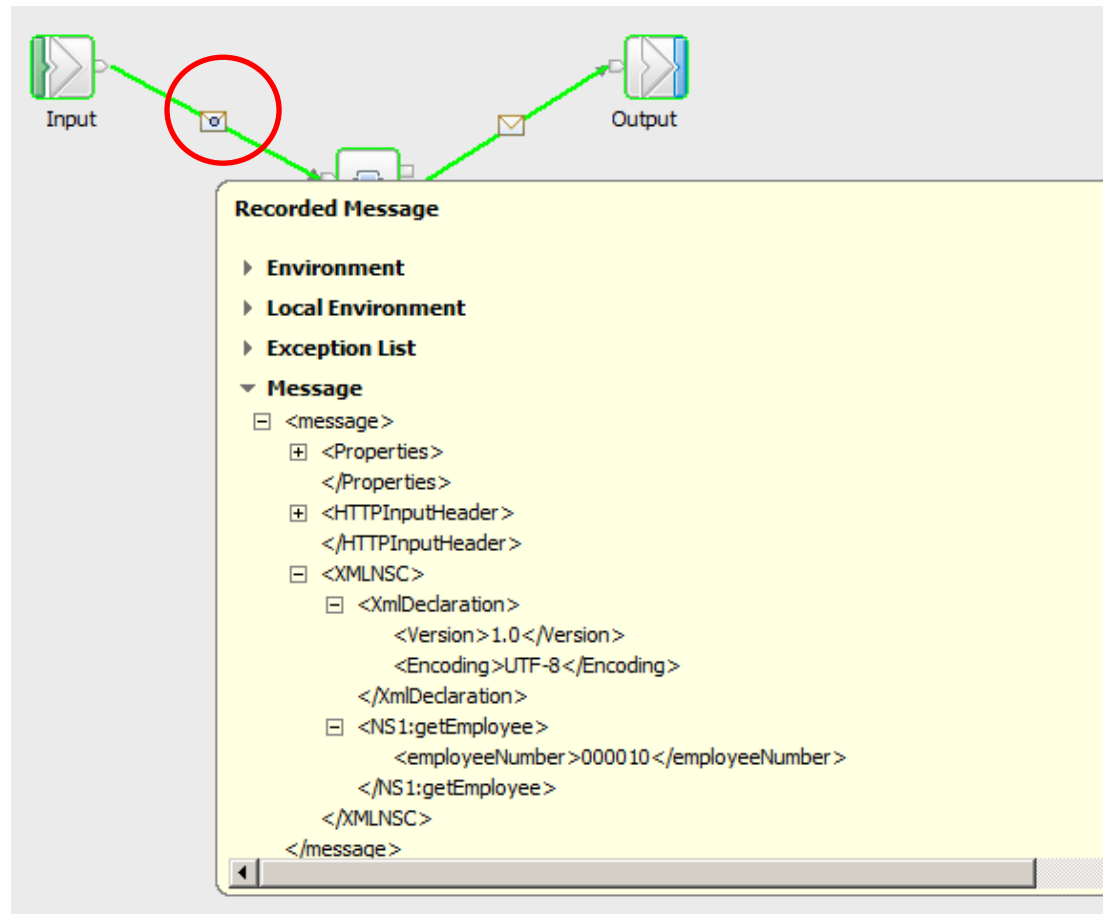


10. Click the icon on the first connector.

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.

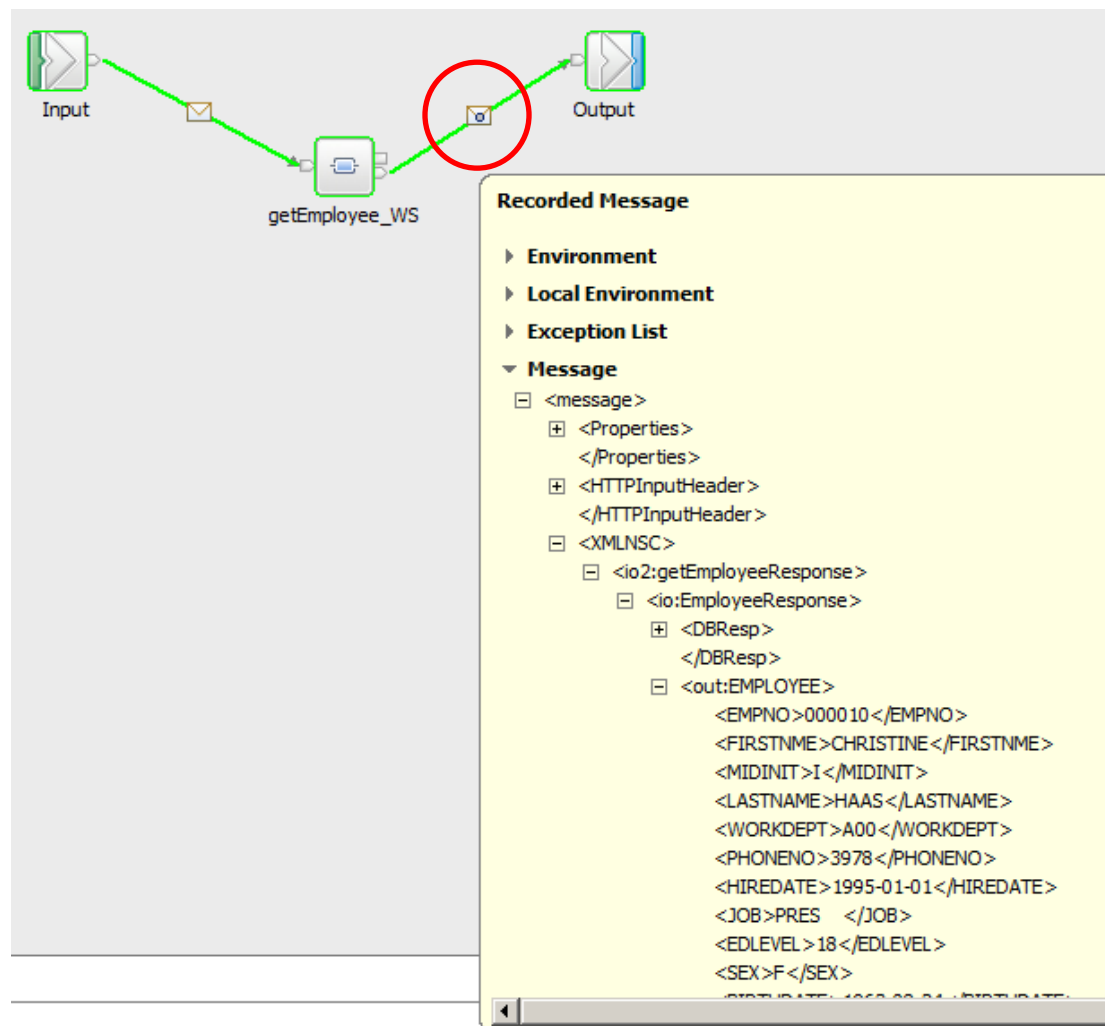
You can expand various parts of the tree, such as LocalEnvironment, Environment, ExceptionList, as well as the Message.

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

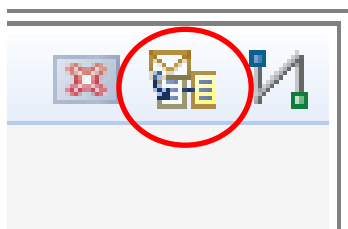


11. Click the icon on the final connector. This time, you will see the message tree after the employee record has been retrieved from the database.

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



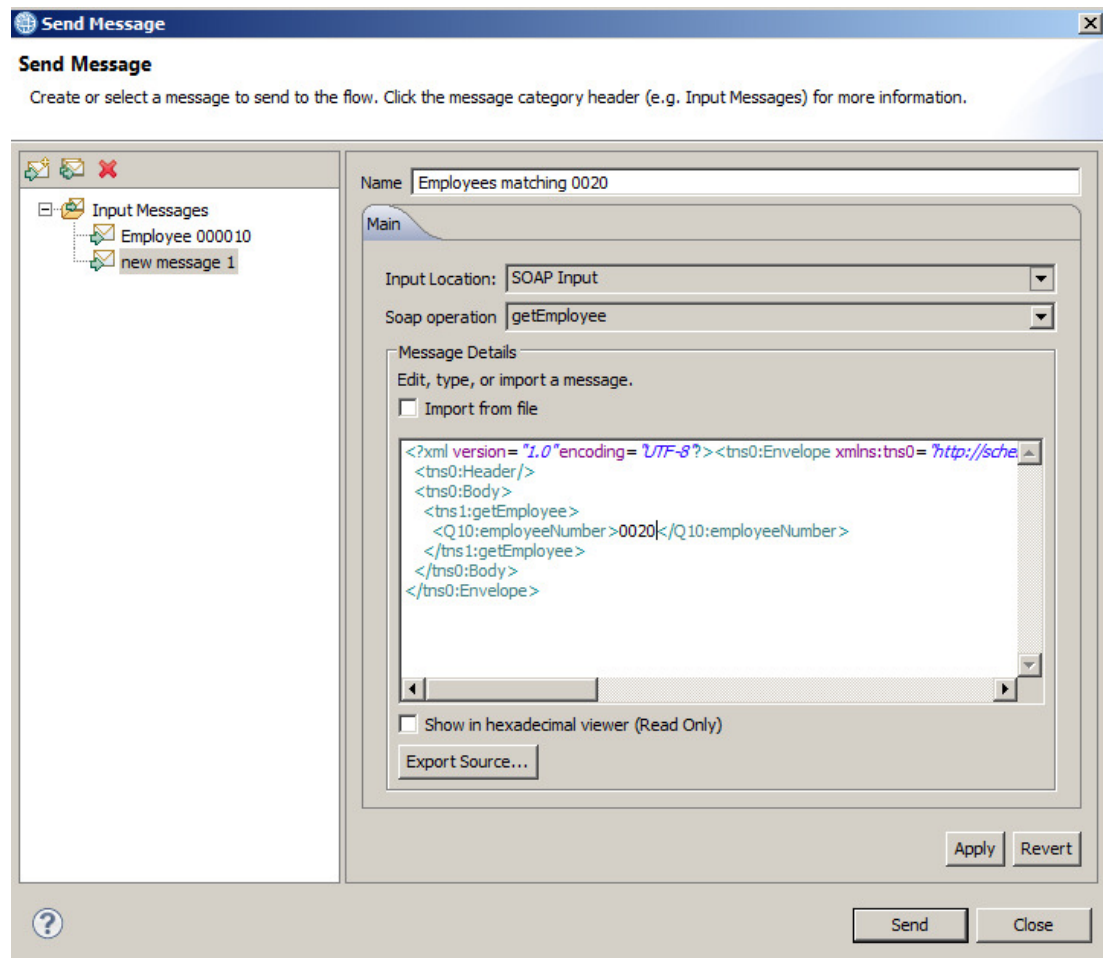
12. Click the Send Message icon again.





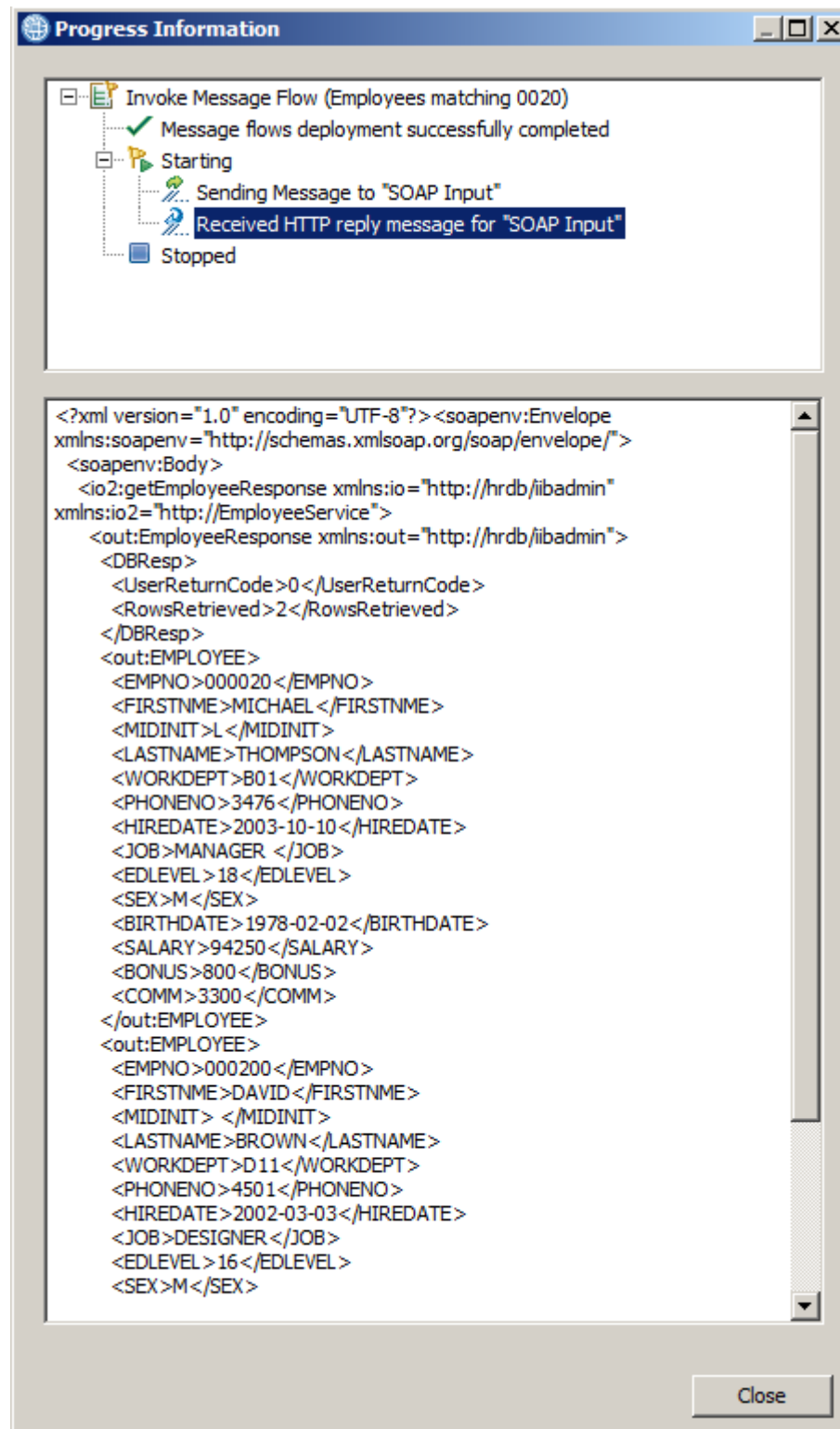
13. Create a new test as before. Name it "Employees matching 0020".

Set the employeeNumber to 0020, and click Send.



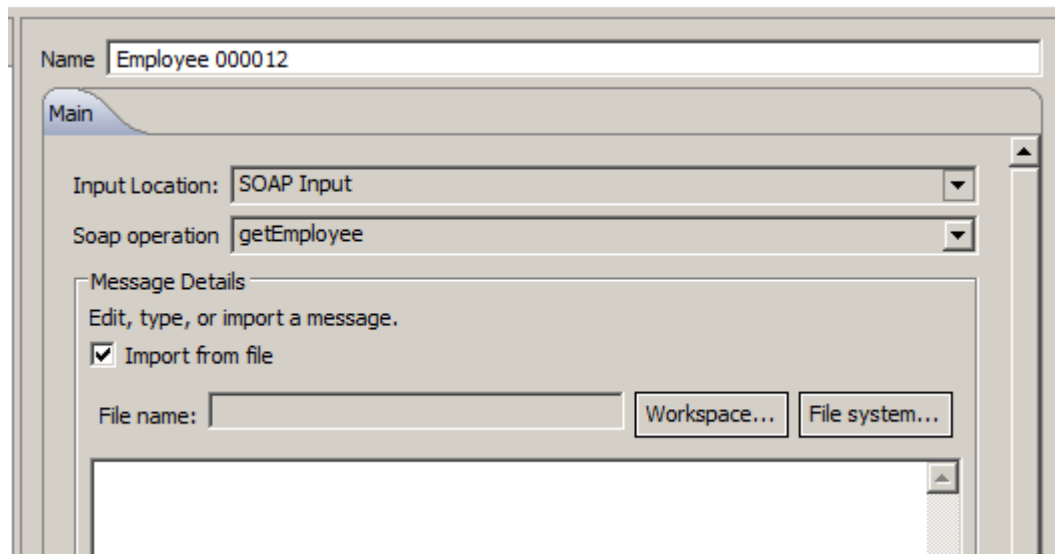
14. The returned message will contain two employees, both containing "0020" in the employee number field.

RowsRetrieved = 2.

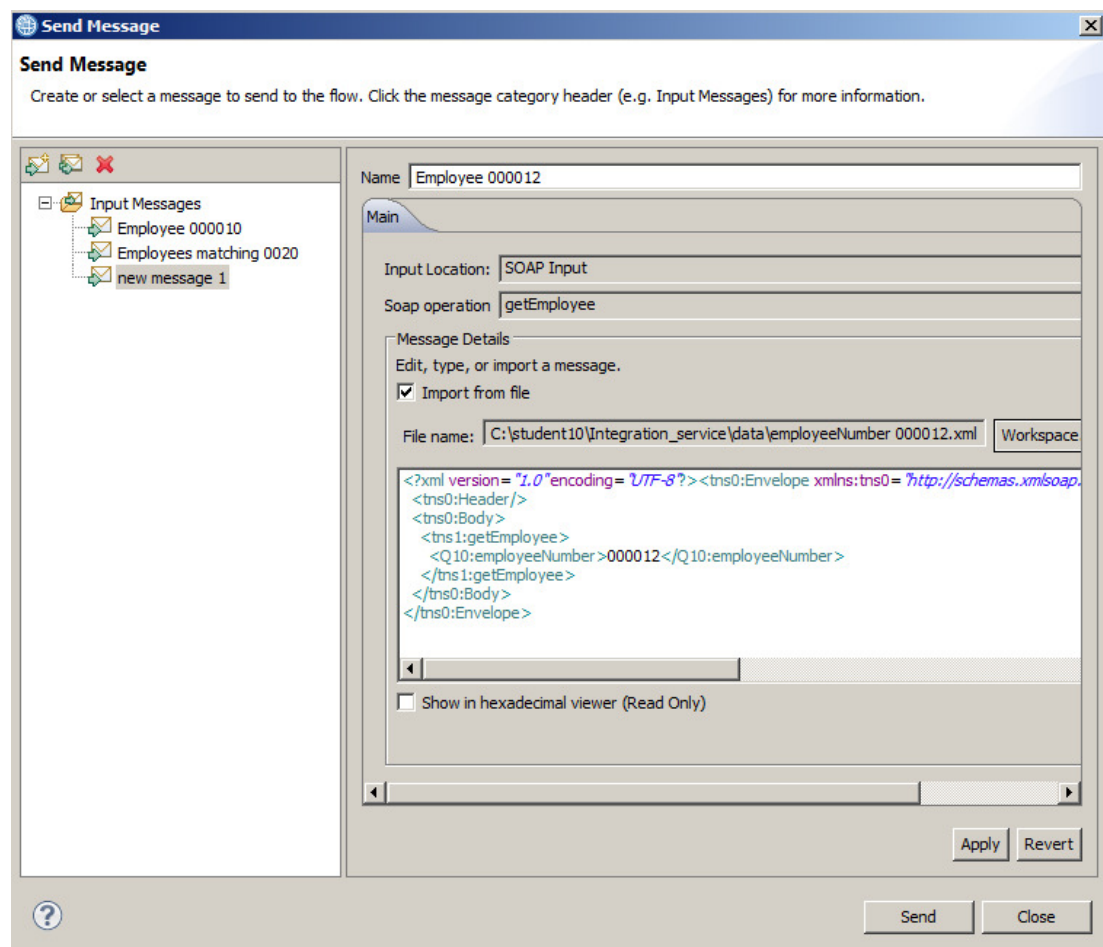


15. Create a new test as before. Name it "Employee 000012".

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

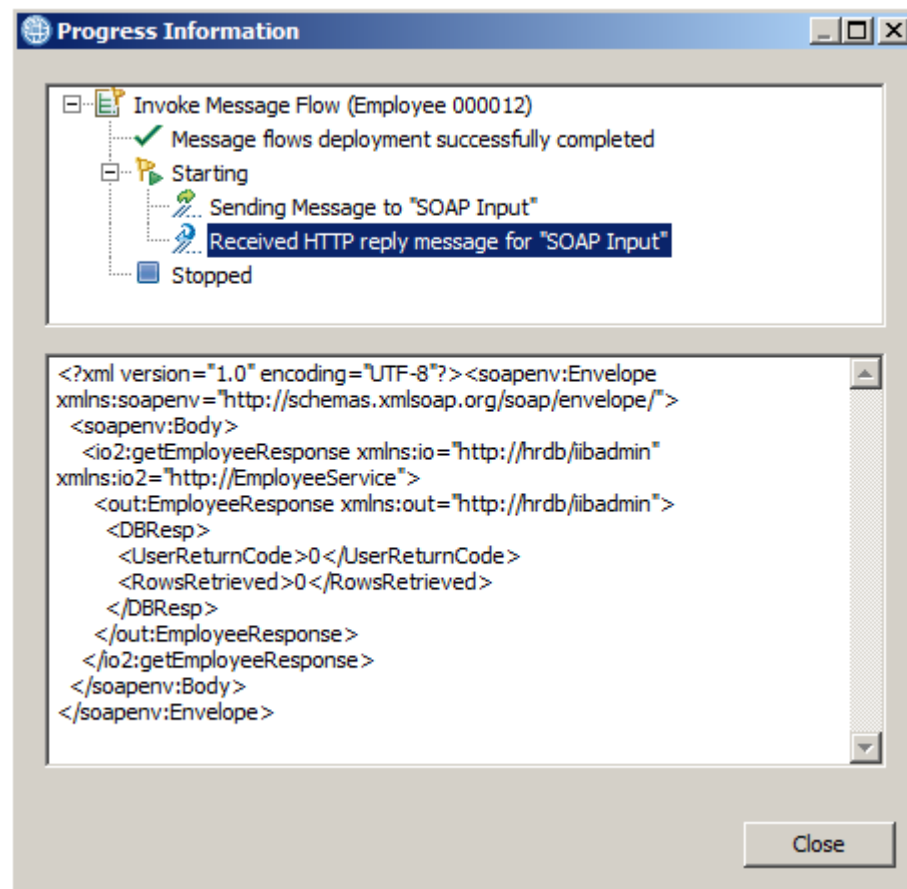


16. Click Send, and use the same tools to review the test outcome.



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

RowsRetrieved = 0.



END OF LAB GUIDE