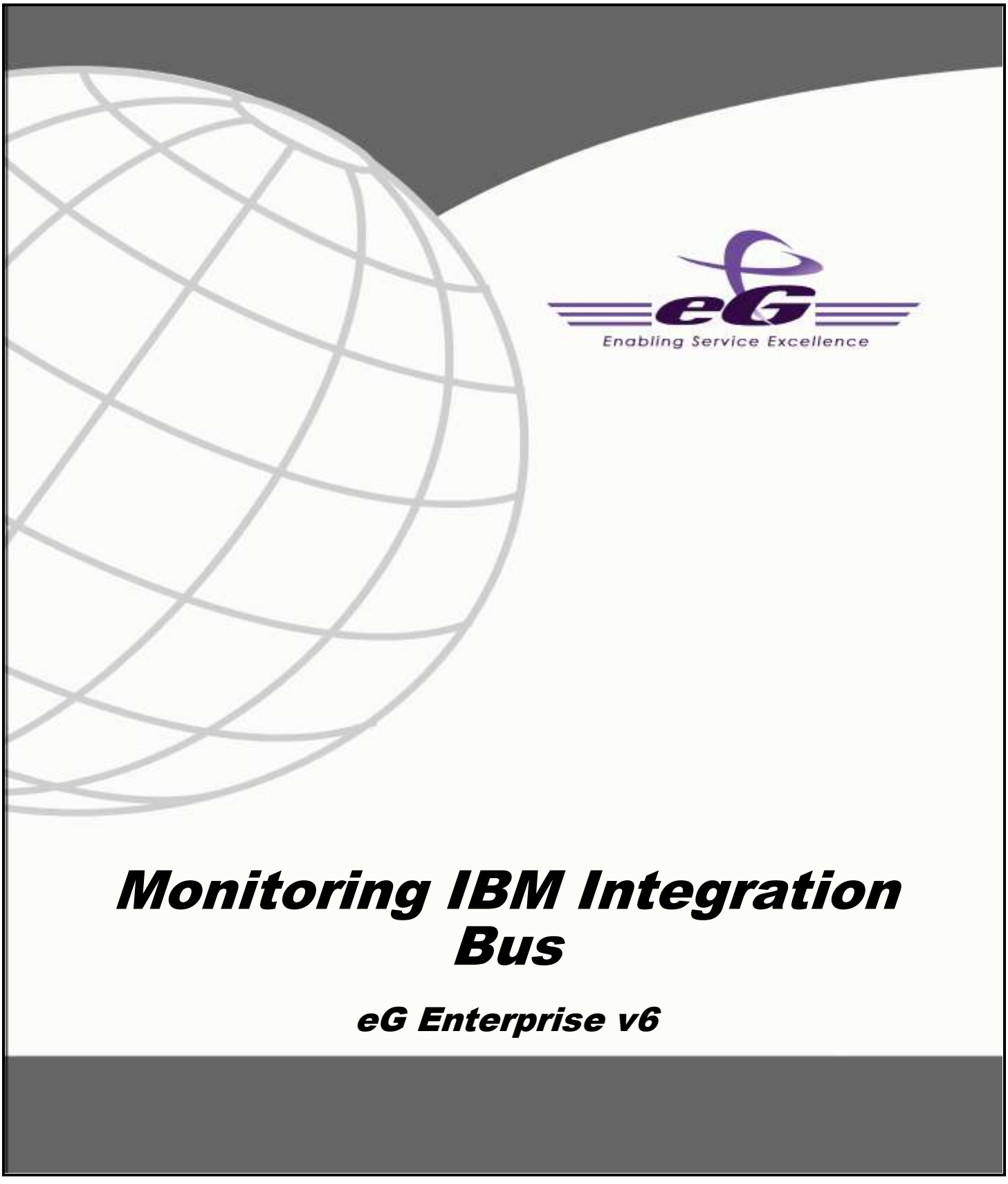

Monitoring IBM Integration Bus

eG Enterprise v6

Restricted Rights Legend

The information contained in this document is confidential and subject to change without notice. No part of this document may be reproduced or disclosed to others without the prior permission of eG Innovations Inc. eG Innovations Inc. makes no warranty of any kind with regard to the software and documentation, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose.

Trademarks

Microsoft Windows, Windows NT, Windows 2000, Windows 2003 and Windows 2008 are either registered trademarks or trademarks of Microsoft Corporation in United States and/or other countries.

The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

Copyright

Table of Contents

Table of Figures

1

Introduction

The IBM® Integration Bus, formerly known as the IBM WebSphere Message Broker Family, provides a variety of options for implementing a universal integration foundation based on an enterprise service bus (ESB). Implementations help to enable connectivity and transformation in heterogeneous IT environments for businesses of any size, in any industry and covering a range of platforms including cloud and z/OS. IBM Integration Bus provides a universal integration capability that addresses a wide range of integration scenarios. These scenarios include web services such as SOAP and REST, messaging, database, file, ERP systems, mobile, physical devices, email, custom systems and more.

IBM® Integration Bus enables information packaged as messages to flow between different business applications, ranging from large traditional systems through to unmanned devices such as sensors on pipelines.

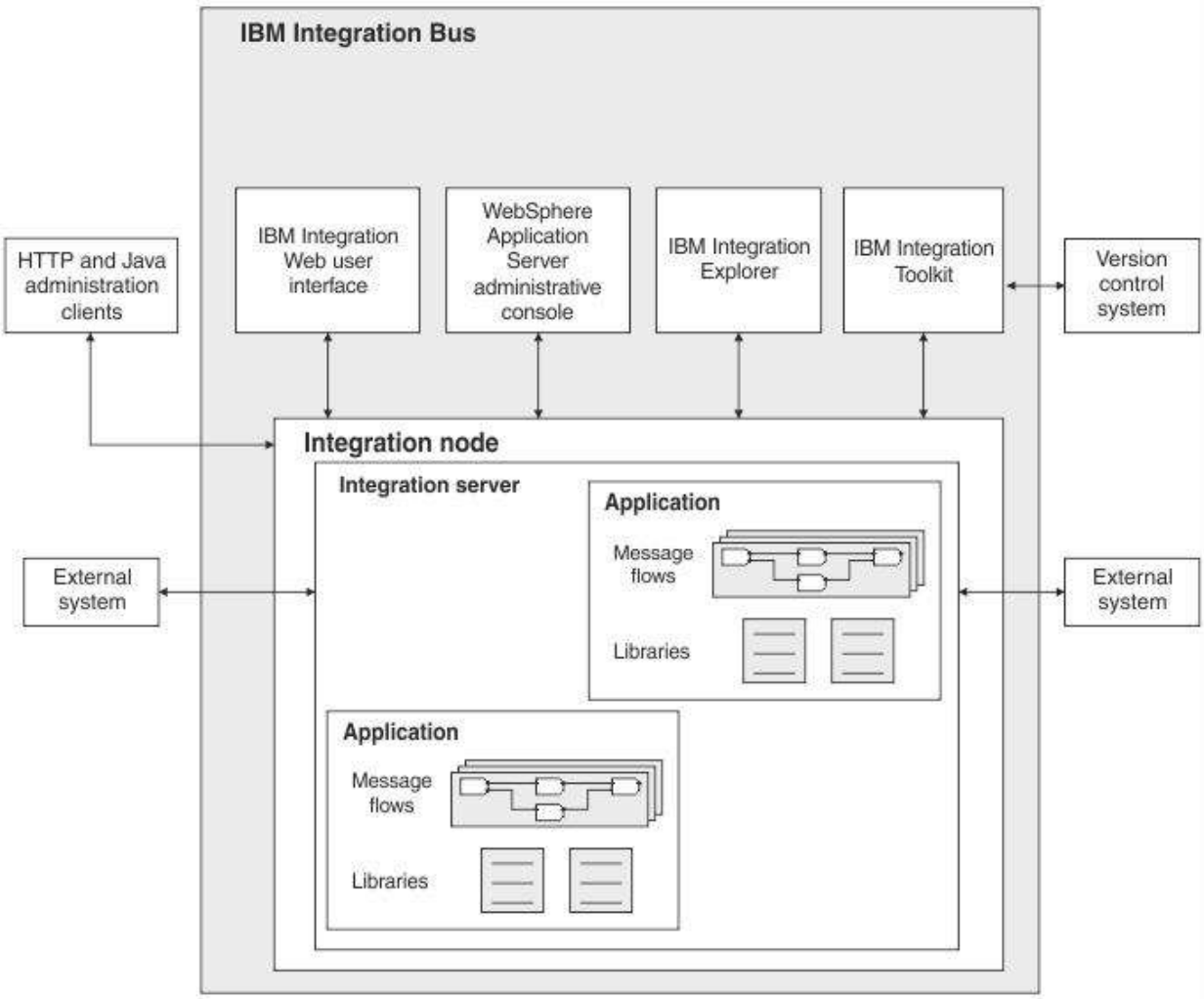

Figure 1: The architectural overview of the IBM Integration Bus

Using the IBM Integration Bus, it is possible to connect applications together, regardless of the message formats or protocols that they support. This connectivity means that your diverse applications can interact and exchange data with other applications in a flexible, dynamic, and extensible infrastructure.

The interactions with the IBM Integration Bus can be broadly classified in to two categories:

Application, development, test and deployment

- Patterns provide reusable solutions that encapsulate a tested approach to solving a common architecture, design, or deployment task in a particular context.
- Application or Execution groups enable message flows within the broker to be grouped together. Each broker contains a default execution group. Additional execution groups can be created as long as they are given unique names within the broker. Each execution group is a separate operating system process and, therefore, the contents of an execution group remain separate from the contents of other execution groups within the same broker. This can be useful for isolating pieces of information for security because the message flows execute in separate address spaces or as unique processes. Message flow applications are deployed to a specific execution group. To enhance performance, the same message flows and message sets can be running in different execution groups.

- Message flows describe your application connectivity logic, which defines the exact path that your data takes in the integration node, and therefore the processing that is applied to it by the message nodes in that flow.
- Message nodes encapsulate required integration logic, which operates on your data when it is processed through your integration node.
- Message trees describe data in an efficient, format independent way. You can examine and modify the contents of message trees in many of the nodes that are provided, and you can supply additional nodes to your own design.
- You can implement transformations by using graphical mapping, Java™, PHP, ESQL, and XSL, and can make your choice based on the skills of your workforce without having to provide retraining.

Operational Management and Performance

- An extensive range of administration and systems management options for developed solutions.
- Support for a wide range of operating system and hardware platforms.
- A scalable, highly performing architecture, based on requirements from traditional transaction processing environments.
- Tight integration with software products, from IBM and other vendors, that provide related management and connectivity services.

In large environments where multiple business applications are integrated with the IBM Integration Bus, administrators may find it difficult to identify a sudden slowdown in the processing rate of the applications and what exactly caused the slowdown. To avoid such erroneous situation, monitoring the IBM Integration Bus becomes imperative. In the forthcoming pages, let us discuss how well the eG Enterprise Suite monitors the IBM Integration Bus in detail.

1.1 How does eG Enterprise Suite monitor the IBM Integration Bus?

The eG Enterprise Suite is capable of monitoring the IBM Integration Bus in both agent based and agentless manner. In order to monitor the IBM Integration Bus, certain pre-requisites need to be fulfilled.

Broad Overview of the Pre-requisites for monitoring the IIB server

- The target IBM Integration Bus server(IIB server) should be running on JRE 1.5. and above.
- Copy the files required for monitoring the IIB server to the **<EG_INSTALL_DIR>/lib** directory as described in Section 1.1.1.
- A JMS Administered Object must be created with a Connection factory and two JMS Destinations. The JMS Destinations stores the messages published by the IIB; the eG agent subscribes to those messages from the JMS Destinations and collects the required metrics for monitoring. The JMS Administered Object can be created using the following approaches:
  - Automatic creation by executing a script
  - Manual creation through the MQ Explorer GUI

In the forthcoming sections, let us discuss the creation of JMS Administered Objects for both agentless and agent based monitoring.

1.1.1 Copying the files required for monitoring the IIB server

To enable the eG agent to monitor an IIB server, certain files need to be copied to the **<EG_INSTALL_DIR>/lib** directory.

1. Ensure that the following jar files are copied from the **[WebSphere MQ install directory/java/lib]** directory to the **<EG_INSTALL_DIR>/lib** directory:
   - com.ibm.mq.commonservices.jar
   - com.ibm.mq.headers.jar
   - com.ibm.mq.jar
   - com.ibm.mq.jmqi.jar
   - com.ibm.mqjms.jar
   - fscontext.jar
   - javax.jms.jar
   - providerutil.jar
   - dhbcore.jar
   - connector.jar
2. Copy the following jar file from the **[WebSphere MQ install directory/java/jre/lib]** directory to the **<EG_INSTALL_DIR>/lib** directory:
   - ibmjsseprovider2.jar
3. Copy the following jar file from the **[IIB install directory/IBM/MQSI/9.0.0.0/classes]** directory(in case of Windows environment) or from the **[IIB install directory/var/MQSI/9.0.0.0/classes]** directory(in case of Linux environment) to the **<EG_INSTALL_DIR>/lib** directory:
   - ConfigManagerProxy.jar
4. If the IIB server is installed in a Windows environment and the eG agent is of 64 bit, copy the following file from the **[Websphere MQ install directory/java/lib64]** directory to the **<EG_INSTALL_DIR>/lib** directory:
   - mqjbnd64.dll

   Alternately, if the eG agent is of 32 bit, copy the following file from the **[Websphere MQ install directory/java/lib64]** directory to the **<EG_INSTALL_DIR>/lib** directory:
   - mqjbnd.dll
5. If the IIB server is installed in a Linux environment, copy the following file from the **[Websphere MQ install directory/java/lib64]** directory (in case of a 64 bit eG agent) or the **[Websphere MQ install directory/java/lib]** directory (in case of a 32 bit eG agent) to the **<EG_INSTALL_DIR>/lib** directory:
   - libmqjbnd.so

After copying all the required files, remember to restart the eG agent. If IIB monitoring is done in an agentless manner, these jar files should be available on the remote agent that will perform the monitoring.

1.1.2 Configuring eG Agent to Monitor Queue Manager using a Custom Channel

Typically, to monitor a queue manager, the eG agent needs to be configured with access to the default server connection channel on WebSphere MQ – i.e., the **SYSTEM.DEF.SVRCONN** channel. In high security environments however, administrators may prefer not to expose this default channel and its access privileges to the eG agent. In such environments therefore, you need to create a custom server connection channel to enable the eG agent to monitor the queue manager. The below section explains in detail about how to create a Custom Channel.

1.1.2.1 Creating a Custom Channel

To create a custom channel for monitoring a Queue manager, do the following:

1. Execute the following command from the bin directory of the MQ Install directory to start the MQ Series commands (MQSC) for the queue manager:

   runmqsc <Queue Manager name>

   For example, if the name of the Queue manager is **IB9QMGR**, then the command will be as follows:

   runmqsc IB9QMGR

2. Once the MQSC is started, you can execute commands to create a custom channel. Say for example, the name of the custom channel is **eGChannel**, execute the commands in the command prompt as follows:

   DEFINE CHANNEL(CLNT.EG.SVRCONN) CHLTYPE(SVRCONN) TRPTYPE(TCP)

3. If you wish to add a description to the custom channel, then you can do so by adding a DESCR section to the above mentioned command. In our example, the description is **Server-Connection channel for eG**. Therefore the command that needs to be executed is as follows:

   DEFINE CHANNEL(CLNT.EG.SVRCONN) CHLTYPE(SVRCONN) TRPTYPE(TCP) DESCR<'Server-Connection channel for eG '>

4. If the command exceeds the stipulated number of characters that need to be typed in the command prompt, then the command can be split by a '+' symbol. In such cases, the command can be executed as follows (see Figure 2):

   DEFINE CHANNEL(CLNT.EG.SVRCONN) CHLTYPE(SVRCONN) TRPTYPE(TCP) +

   DESCR('Server-Connection channel for eG')

5. The custom WebSphere MQ channel will now be created. For a custom channel to take effect in the WebSphere MQ server, either the security cache or the queue manager of the server needs to be refreshed. If the queue manager is refreshed, the current messages of the server may get deleted. Therefore, refresh the security cache by issuing the following command (see Figure 2):

   REFRESH SECURITY(*)

6. Once the security cache is refreshed successfully, you can stop the MQ Series commands by issuing the following command:

   End

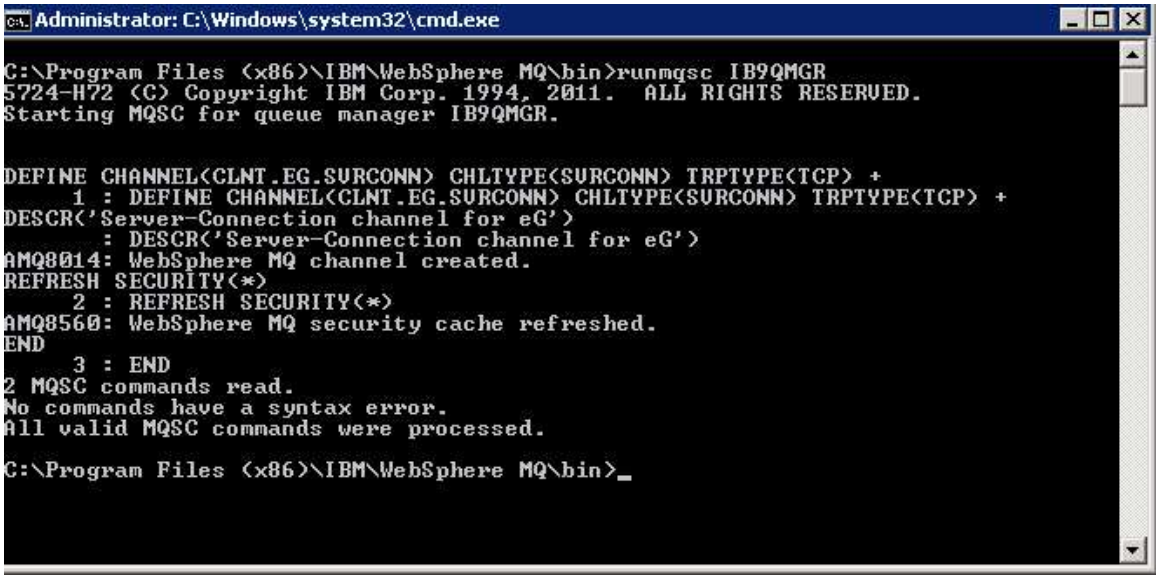

Figure 2: Creating a Custom Channel

The Custom Channel through which the target IIB server is to be monitored is ready.

1.1.3 Agent based monitoring of the IIB server

Let us now discuss on the steps for creating a JMS Administered Object automatically and manually in detail in the forthcoming sections.

1.1.3.1 Automatically creating a JMS Administered Object using scripts

Follow the steps mentioned below to automatically create a JMS Administered Object:

1. By default, the following files will be available in the **<eG_INSTALL_DIR>\lib** folder say for e.g., **C:\egurkha\lib** of the target environment:
   - eGJMSIIBCreateConfig.bat
   - eGJMSIIBCreateConfig.sh
2. If IIB server is installed on a Windows environment, execute the **eGJMSIIBCreateConfig.bat** file from the command prompt of the IIB server host as follows:

   ```
   C:\egurkha\lib>eGJMSIIBCreateConfig.bat
   ```

   Then proceed to step 5.
3. If IIB server is installed on a Linux environment, execute the **eGJMSIIBCreateConfig.sh** file from the shell script.

   ```
   /opt/egurkha/lib>eGJMSIIBCreateConfig.sh
   ```

4. Now, specify the exact location on which the MQ Java installation directory is installed. In our example, the installation path is **/opt/mqm/java/**.

   ```
   Please enter the MQ java installed path: /opt/mqm/java/
   ```

5. Once the respective files for the Windows and Linux environments are executed successfully, you will be asked to provide the input parameters that are required for creating the JMS Administered Object. To begin with,

specify the location of the **JNDI namespace** as mentioned below:

```
Where is the JNDI namespace located?
1. File System
2. LDAP Server
Enter your choice [1]: 1
```

6. Selecting the **File System** option indicates that the **JNDI namespace** is located on the IIB server host itself. Once you have chosen the **File System** option, you will be required to provide the IP address of the IIB server host.

```
Enter the host IP address on which the IBM Integration Bus is installed:
192.168.9.165
```

7. To create the **Connection Factories** and the **Destinations**, specify the name of the queue manager.

```
Enter the name of the QManager: IB9QMGR
```

8. Now, specify the name of the node in the IIB server host that is to be monitored as shown below:

```
Enter the name of the Integration node: IB9NODE
```

9. Finally, specifying the port on which the queue manager listens will automatically create the **JMS Administered Object** along with the **Connection Factories** and the **Destinations**.

```
Enter the port on which the QManager listens: 2414
```

10. If all the input parameters are successfully validated in the target Windows environment, then the following details will appear:

```
********************************************************************************
Following are the parameters that need to be provided while configuring the tests:

HOST = 192.168.9.165
QMANAGER NAME = IB9QMGR
QMANAGER PORT = 2414
INTEGRATION NODE NAME = IB9NODE
JNDI NAMESPACE LOCATION = File
SSL = No
USERDN ==> none
JNDI PROVIDER URL ==> file:///C:/eGurkha/lib/IIB
TOPICCONNECTIONFACTORY = eG_TCF_192.168.9.165_2414
JMS RESOURCE STATS = eG_Res_Topic_192.168.9.165_2414
JMS FLOW STATS = eG_Msg_Topic_192.168.9.165_2414
********************************************************************************
```

11. If your target environment is Linux, then the following details will appear after the successful validation of the input parameters:

```
********************************************************************************
Following are the parameters that need to be provided while configuring the tests:

HOST = 192.168.9.165
QMANAGER NAME = IB9QMGR
QMANAGER PORT = 2414
INTEGRATION NODE NAME = IB9NODE
JNDI NAMESPACE LOCATION = File
SSL = No
USERDN ==> none
```

```
JNDI PROVIDER URL ==> file:/opt/egurkha/lib/IIB
TOPICCONNECTIONFACTORY = eG_TCF_192.168.9.165_2414
JMS RESOURCE STATS = eG_Res_Topic_192.168.9.165_2414
JMS FLOW STATS = eG_Msg_Topic_192.168.9.165_2414
**************************************************************************
```

12. Specify the values mentioned above against the appropriate fields while configuring the tests for the IIB server that is to be monitored. To know how to configure the tests, refer to Chapter 2 of this document.

If you fail to notice the details mentioned in Step 10 and Step 11, then you can figure them out in the **<EG_INSTALL_DIR>\lib\IIB\eG_IIB_test_config.log** file. If multiple IIB nodes are monitored in the target environment, then the test parameter values will be stored for each IIB node separately.

1.1.3.2 Manually creating a JMS Administered Object in the IIB server using MQ Explorer console/GUI

Prior to creating a JMS Administered Object, do the following:

1. Login to the IIB server host.
2. In the location of your choice, say for e.g., **C:**, create a new empty folder. In our example, the name of the folder is **JNDI-Directory**. This folder, otherwise called the JNDI namespace is used to store the contents of the JMS Administered Object that you are about to create.

Follow the steps mentioned below to create a JMS Administered Object:

1. From the IIB server host, follow the menu sequence: *Start -> All Programs -> IBM Integration Bus 9.0 -> IBM Integration Explorer* (see Figure 3).

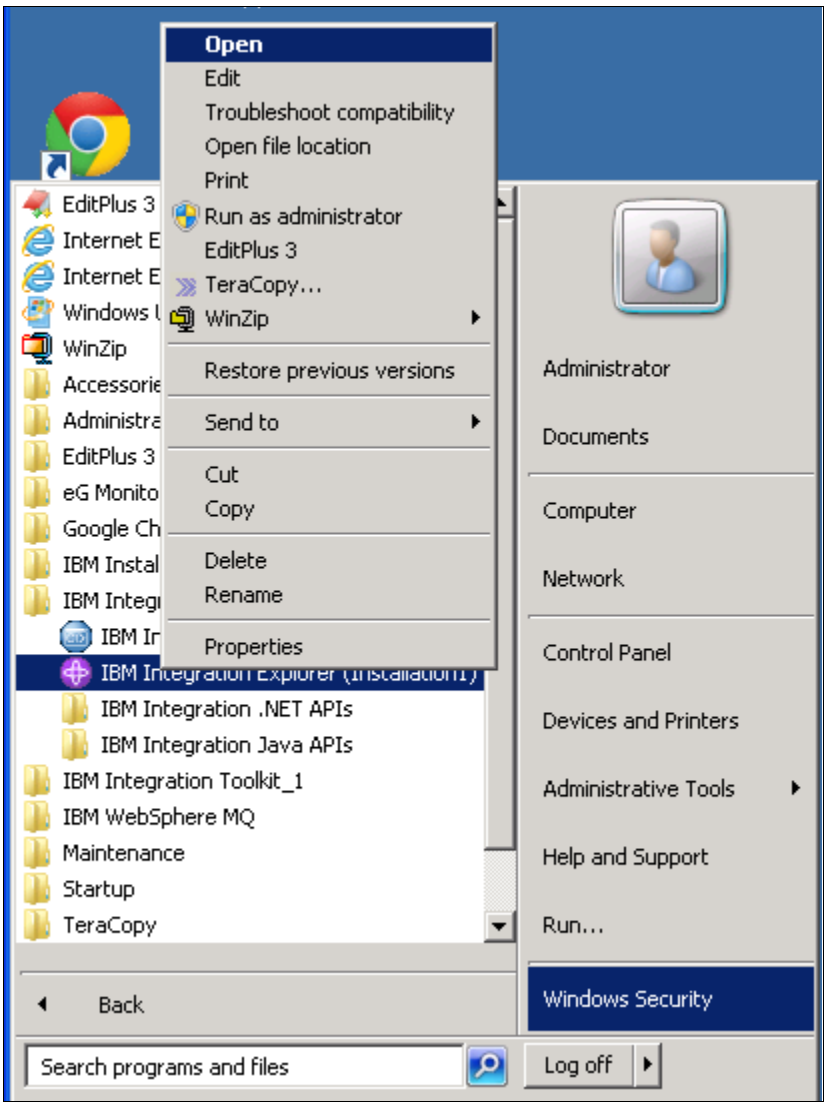

Figure 3: Opening the IBM Integration Explorer menu

2. Open the **IBM Integration Explorer** as shown in Figure 3. Figure 4 will then appear.

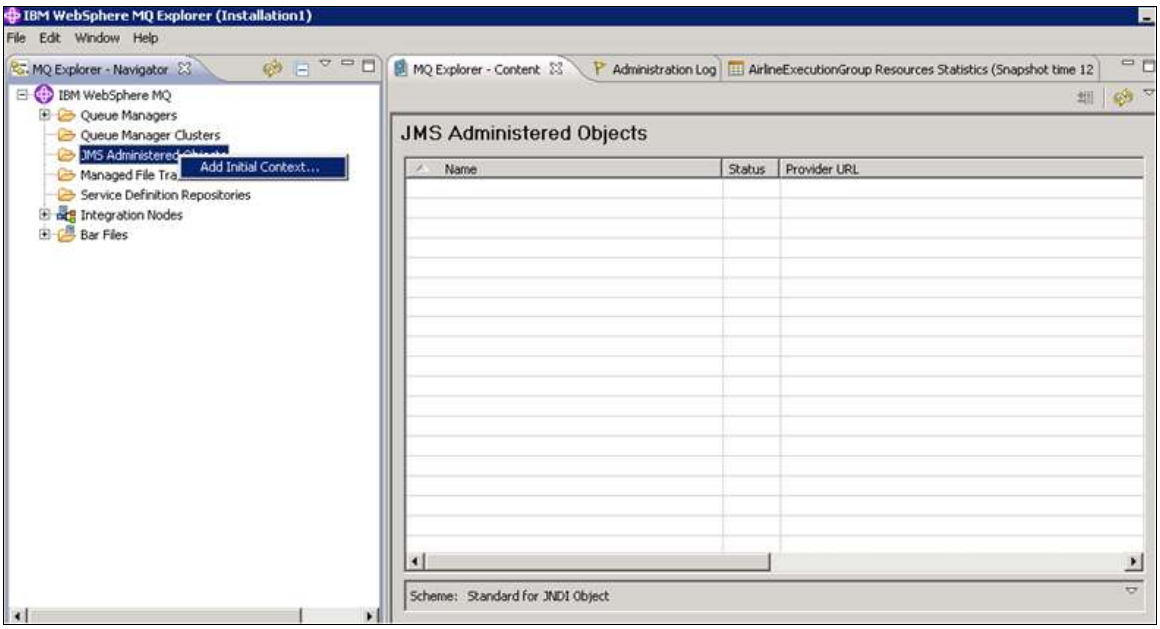

Figure 4: Opening the IBM WebSphere MQ Explorer

3. Right clicking the **JMS Administered Objects** node from the **MQ Explorer – Navigator** panel will display the **Add Initial Context** option (see Figure 4). An initial context is the root of the JNDI namespace that is used to access the JMS objects that are stored in them.

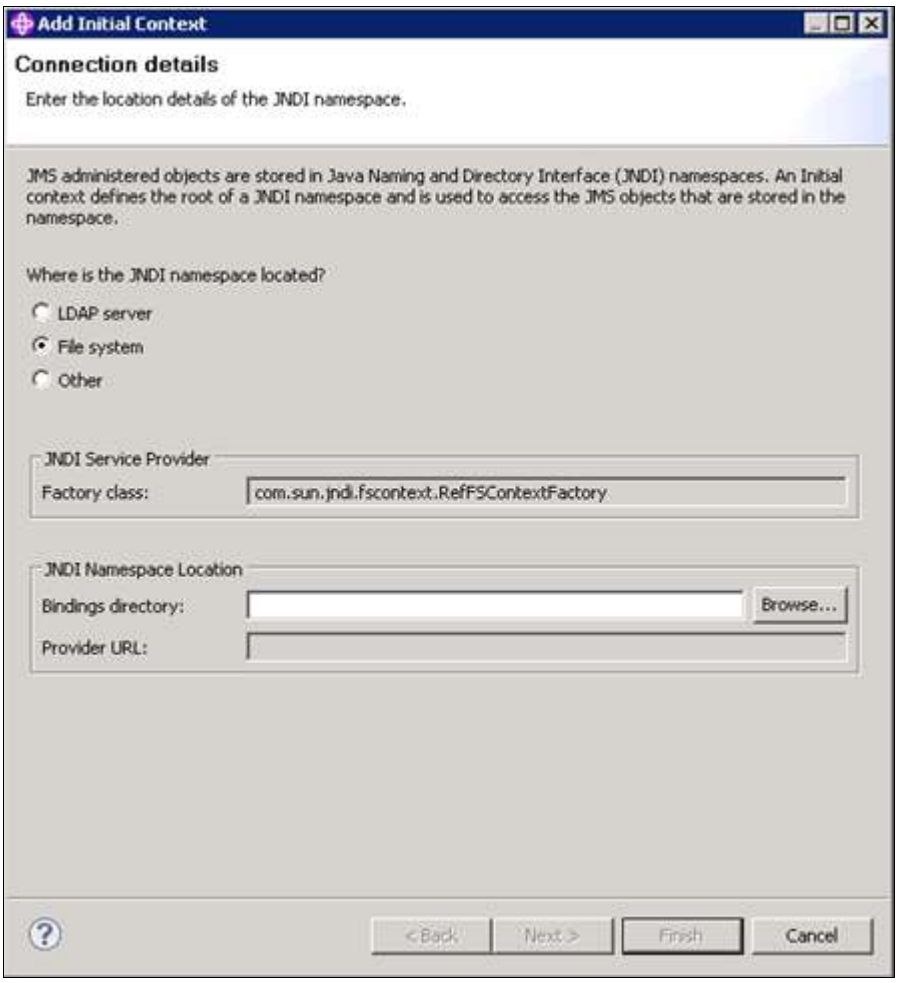

Figure 5: Choosing the location of the JNDI namespace

4. In Figure 5 that appears, you will be required to provide the location of the JNDI namespace that you have created for storing the JMS Administered Object. If you have created the JNDI namespace in the IIB server host itself, then set the **Where is the JNDI namespace located?** to **File system**. If JNDI namespace is not created in the IIB server host, then select the **LDAP server** option.

Figure 6: Selecting the exact path to the created JNDI namespace

5. Select the exact path to the JNDI namespace (see Figure 6) that you have created by clicking the **Browse** button in Figure 5. This will automatically populate the **Bindings directory** and the **Provider URL** text boxes (see Figure 7).

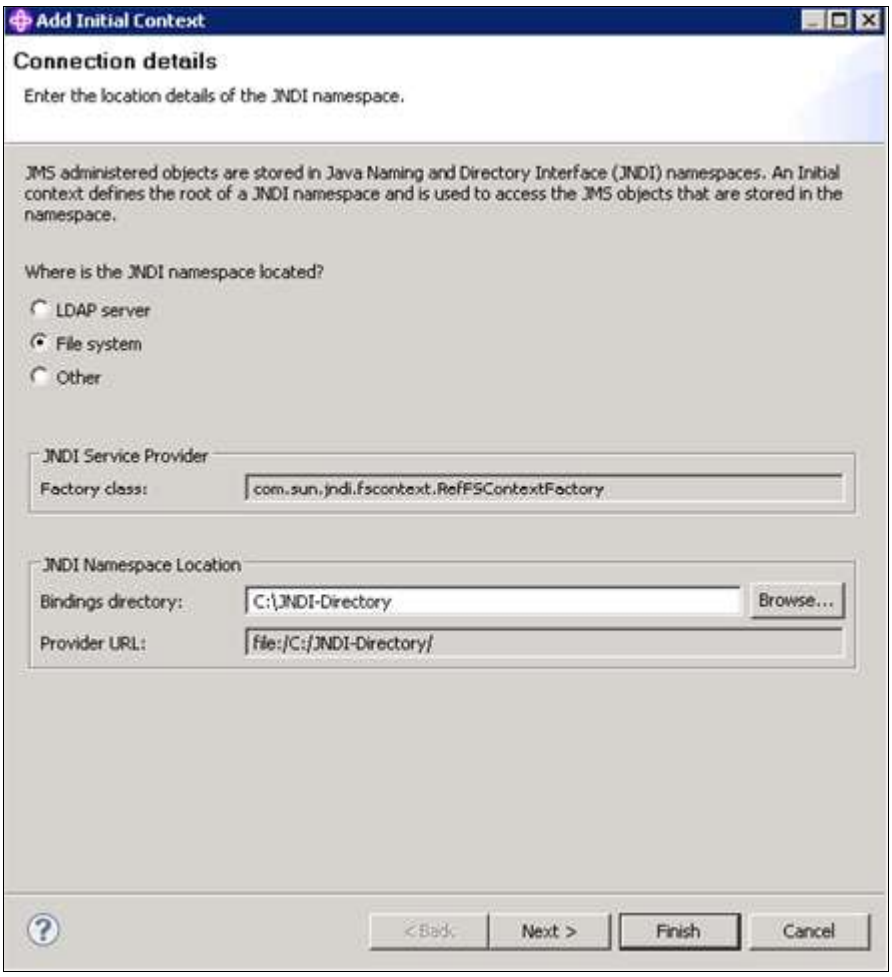

Figure 7: Specifying the exact path of the JNDI namespace

6. Clicking the **Next** button in Figure 7 will navigate you to Figure 8. Provide a name of your choice in the **Context nickname** textbox.

Figure 8: Providing a nickname for the JMS Administered Object to be created

7. Clicking the **Finish** button in Figure 8 will complete the initial context specification. The nickname specified in Figure 8 will be displayed as the name of the **JMS Administered object** in the **MQ Explorer – Navigator** section. In our example, the nickname is not specified and the **Provider URL** is automatically displayed as the **JMS Administered Object** with two default folders named **Connection Factories** and **Destinations** (see Figure 9).

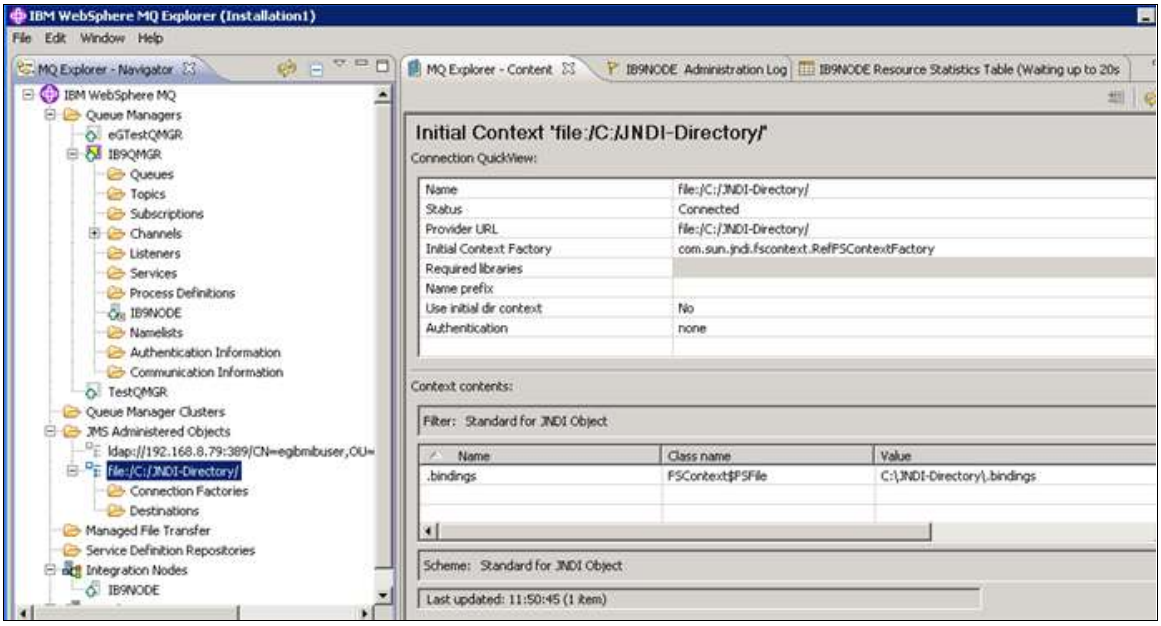

Figure 9: Viewing the created JMS Administered Object

Now let us discuss the steps that need to be followed for creating a connection factory and the destinations:

1.1.3.2.1 Creating a Connection Factory

For an eG agent to connect to the IIB and collect the desired metrics, a connection needs to be established between the eG agent and the Queue manager of the IIB that is to be monitored. To establish such a connection, you would be required to create a connection factory. Follow the below mentioned steps to create a connection factory:

1. Right click **the Connection Factories** folder of Figure 9 and follow the menu sequence: *New -> Connection Factory* (see Figure 10).

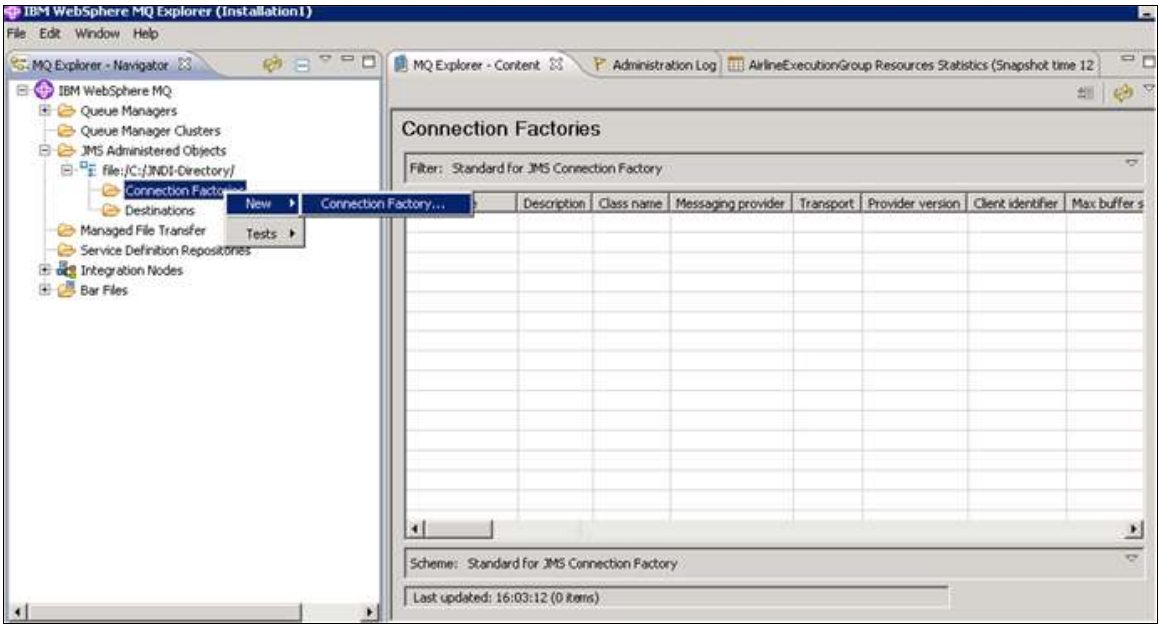

Figure 10: Creating a new connection factory

2. In Figure 11 that appears, specify the **Name** of the connection factory. In our example, the name of the connection factory is **EgTopicConnectionFactory**.

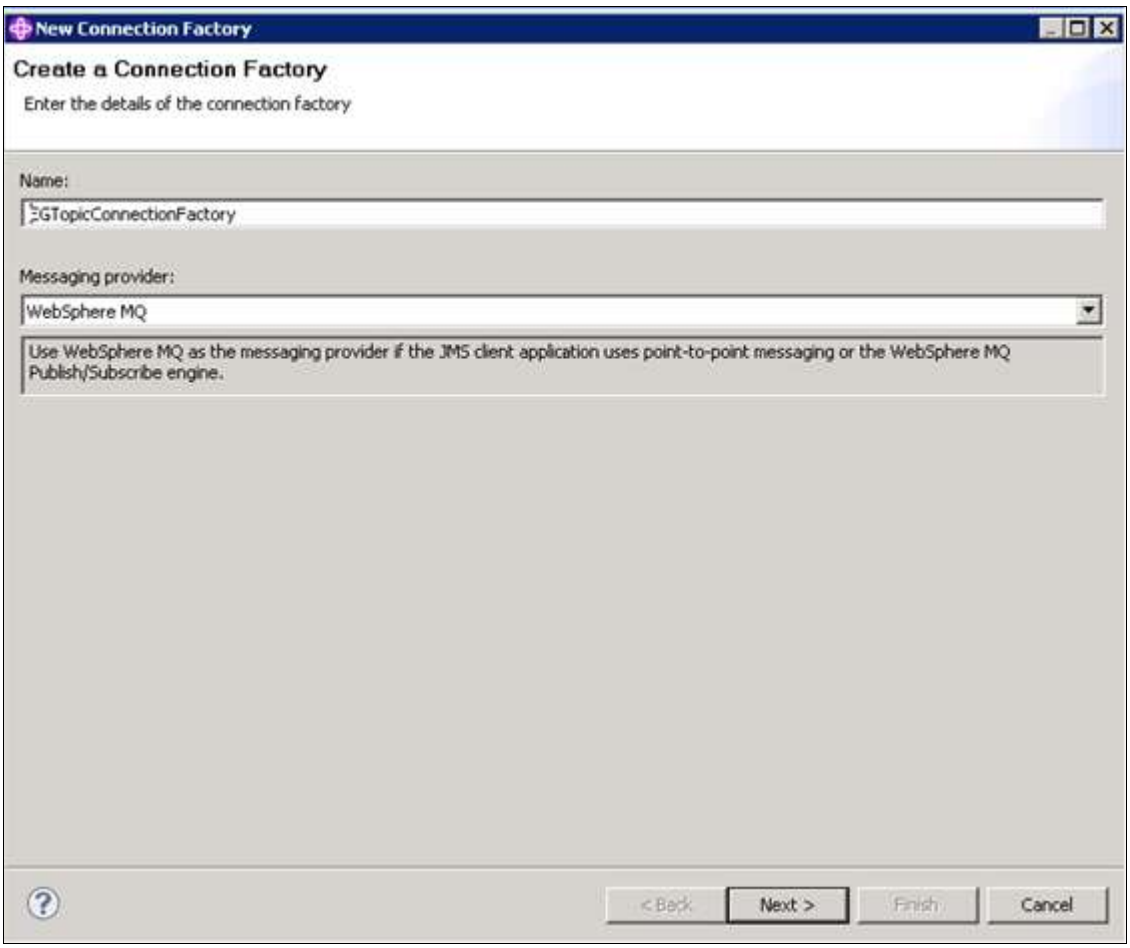

Figure 11: Specifying the name of the connection factory

3. By default, the messages from the IIB are published on the WebSphere MQ. The eG agent is required to subscribe to the messages in order to monitor the IIB. Select **WebSphere MQ** as the **Messaging provider** so that the eG agent can communicate with the WebSphere MQ to collect the required metrics while monitoring the IIB.

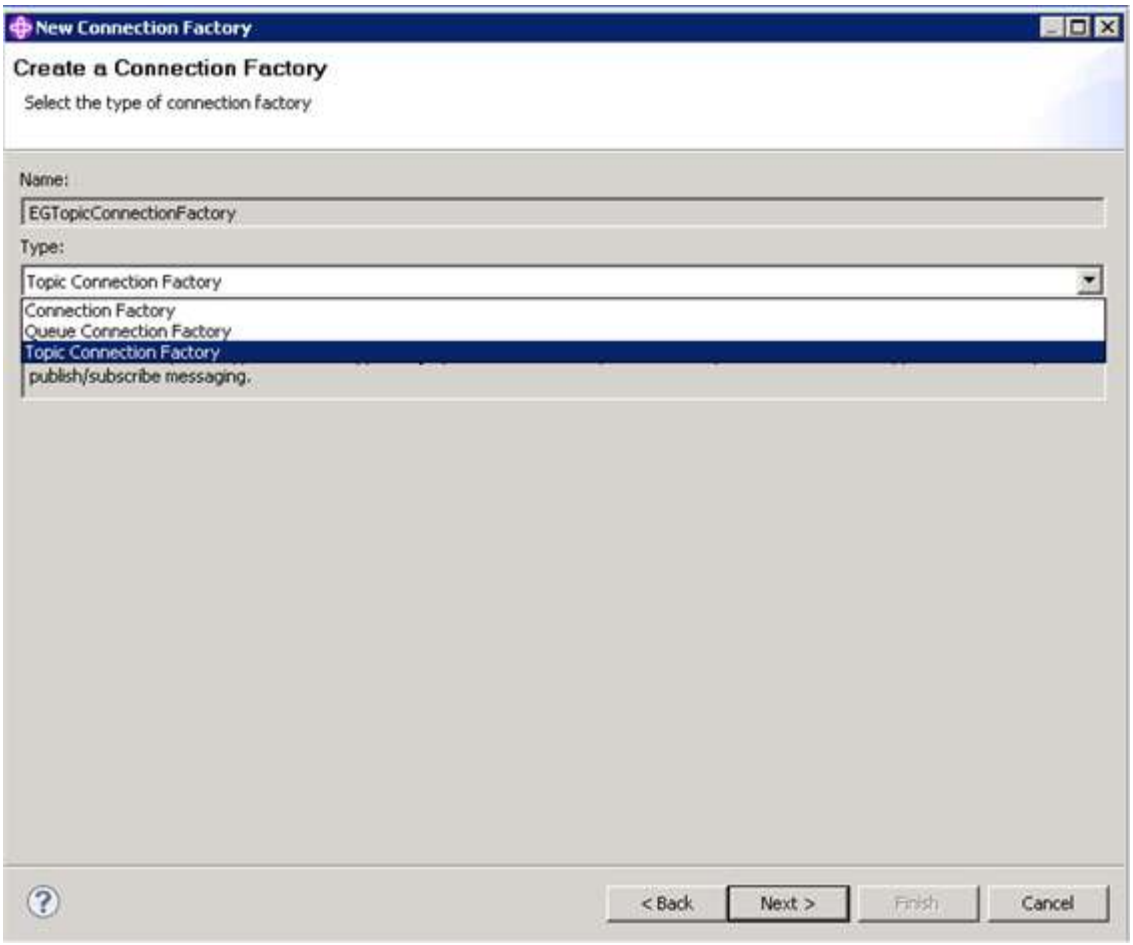

Figure 12: Selecting the type of the connection factory

4. Clicking the **Next** button in Figure 11 will enable you to select the **Type** of the connection factory. Since the eG agent subscribes to the published messages to collect the metrics, select **Topic Connection Factory** option as the **Type**. Clicking the **Next** button will navigate you to Figure 12.

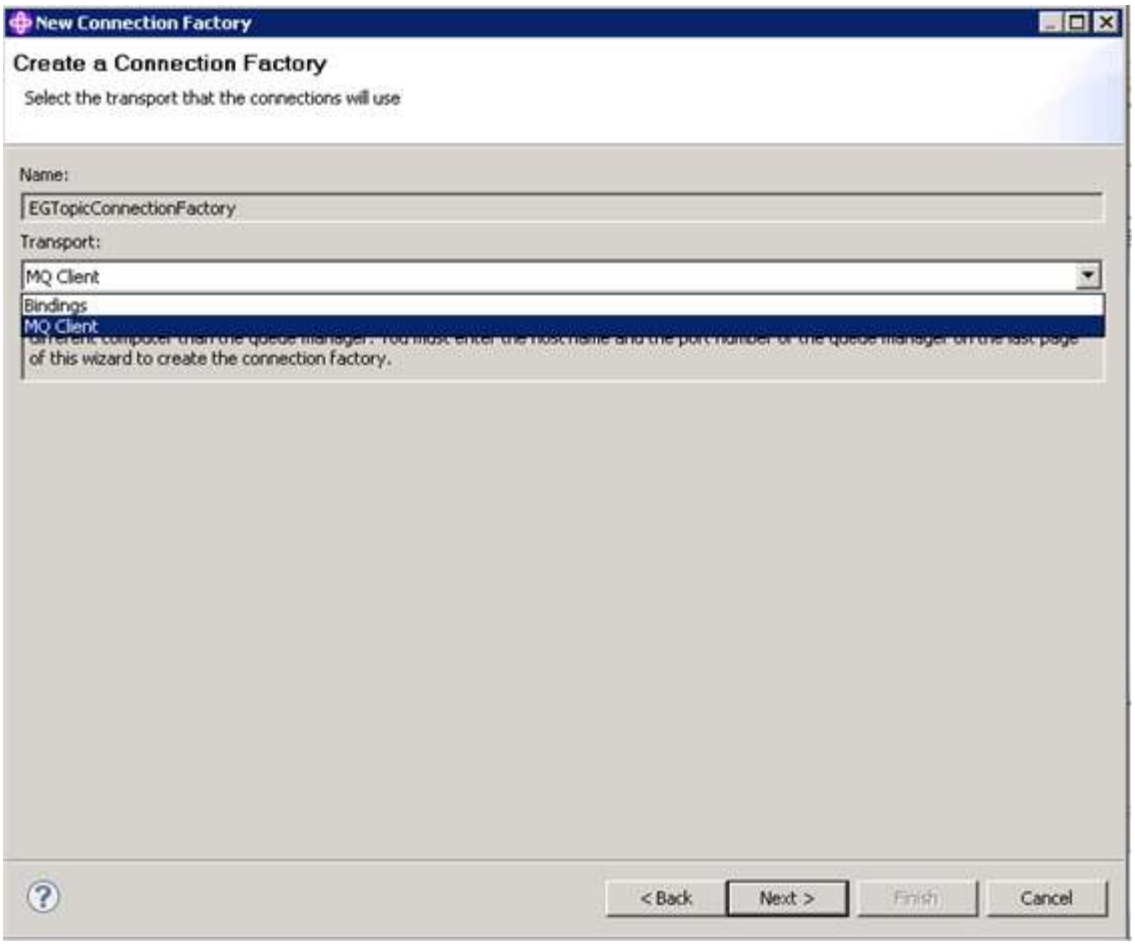

Figure 13: Selecting the transport type of the connection factory

5. Select MQ client as the **Transport** type from Figure 13 and click the **Next** button.

6. The **Change Properties** page will then appear displaying the general settings of the connection factory. Click the **Connection** option in the left panel to select the base queue manager of the IIB node that is to be monitored (see Figure 14). Clicking the **Select**... button will list out all the available Queue managers in the **Select the Base Queue manager** pop up window. Select the queue manager of your choice and click the **OK** button in Figure 14.

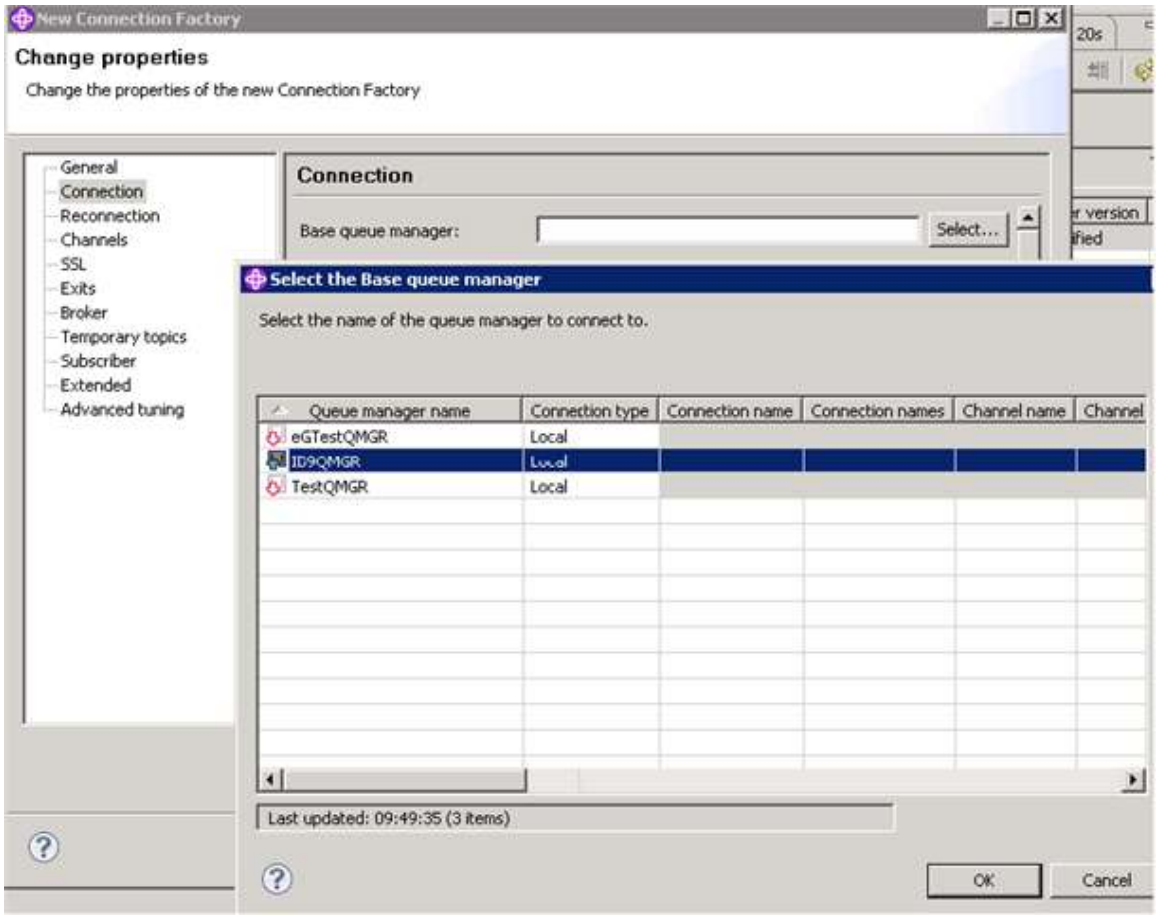

Figure 14: Selecting the Base queue manager

7. In the **Connection list** of Figure 15, specify the host name i.e., the IP address of the chosen queue manager and the port on which the queue manager listens in the following format: **IP address(Port)**.

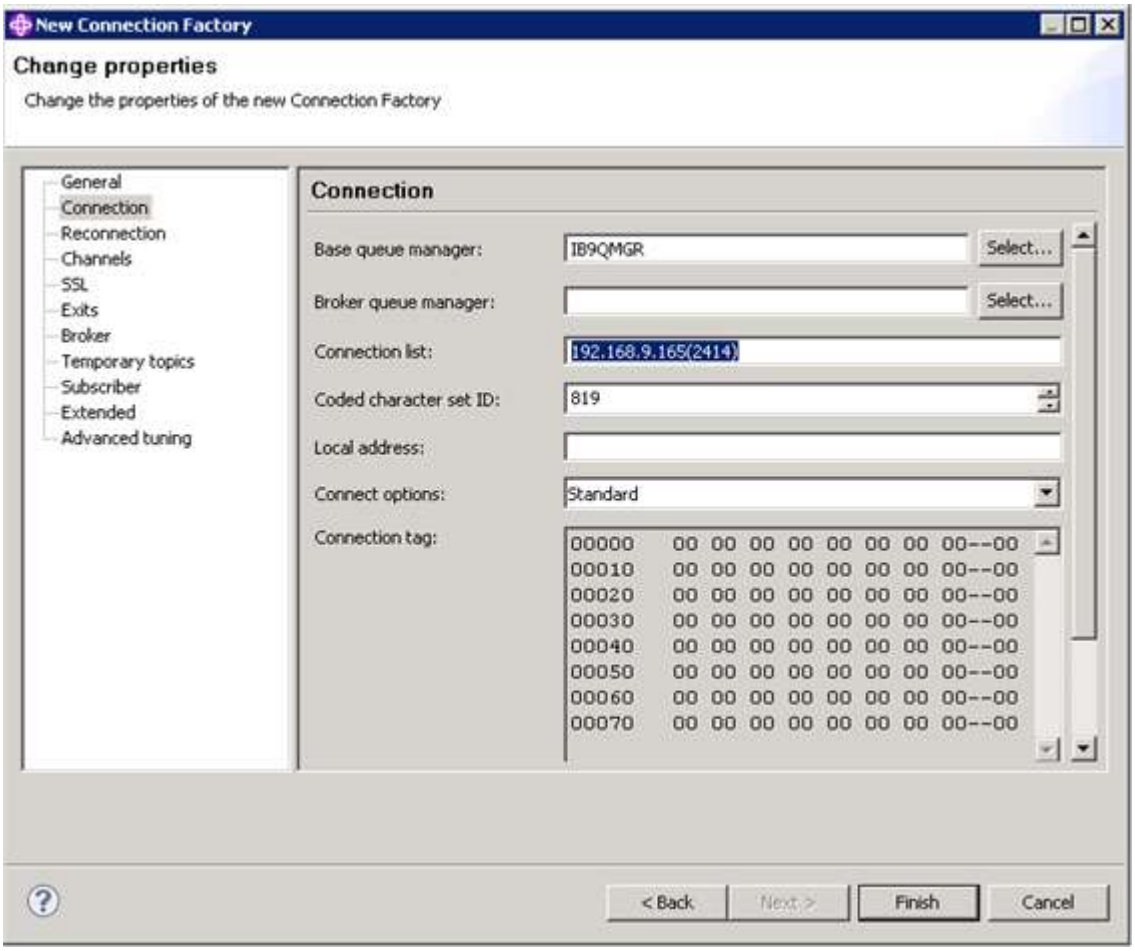

Figure 15: Specifying the host name and port number of the chosen queue manager

8. If you have chosen to use a custom server connection channel for monitoring by the eG Enterprise Suite, then you may be required to select the server connection channel of your choice from the **Channels** option of the **Change Properties** page. The default server connection channel will be listed in the **Channel** text box as shown in Figure 16.

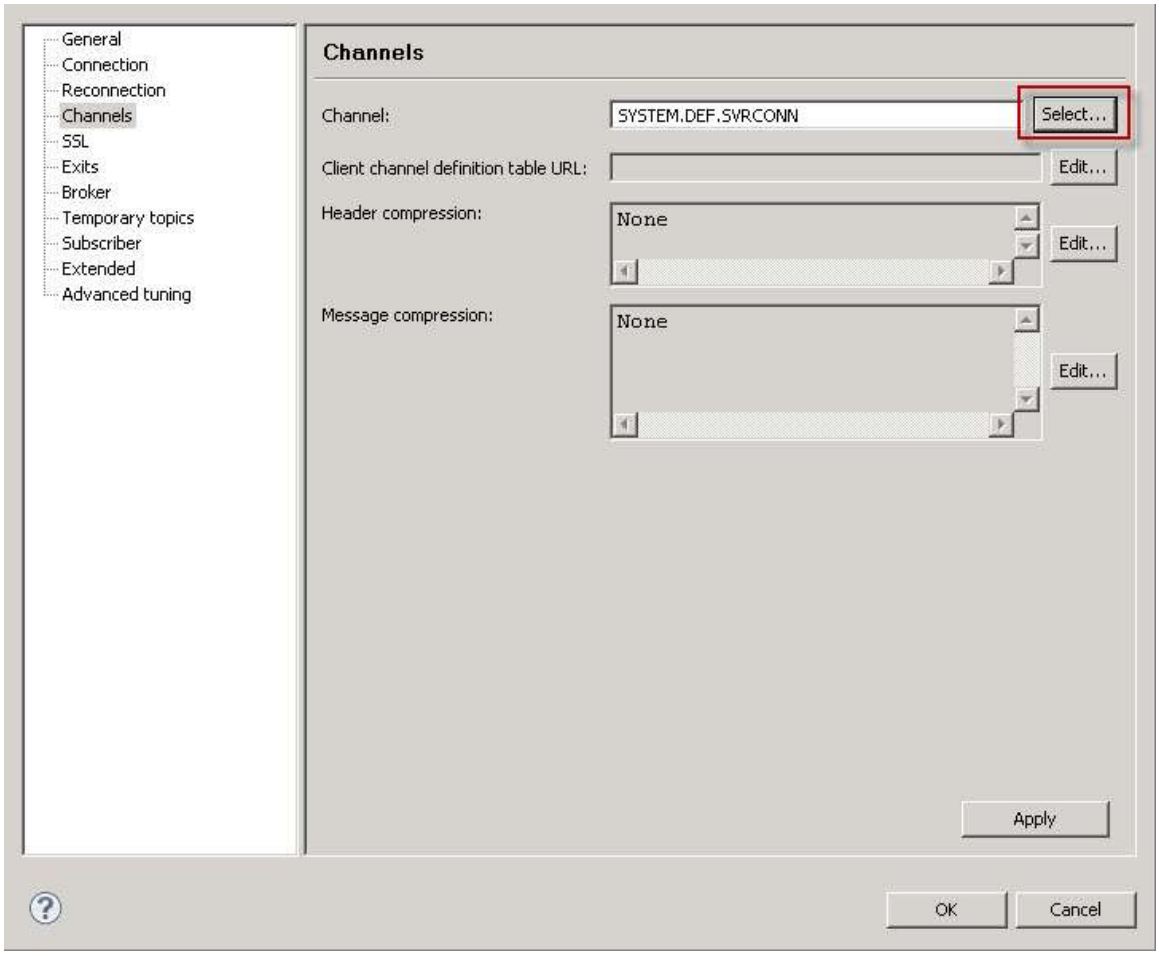

Figure 16: The default server connection channel

9. Clicking the **Select** button against the **Channel** text box in Figure 16 will lead you to Figure 17 where you will be required to choose the server connection channel that you have created for monitoring purpose by the eG Enterprise Suite.

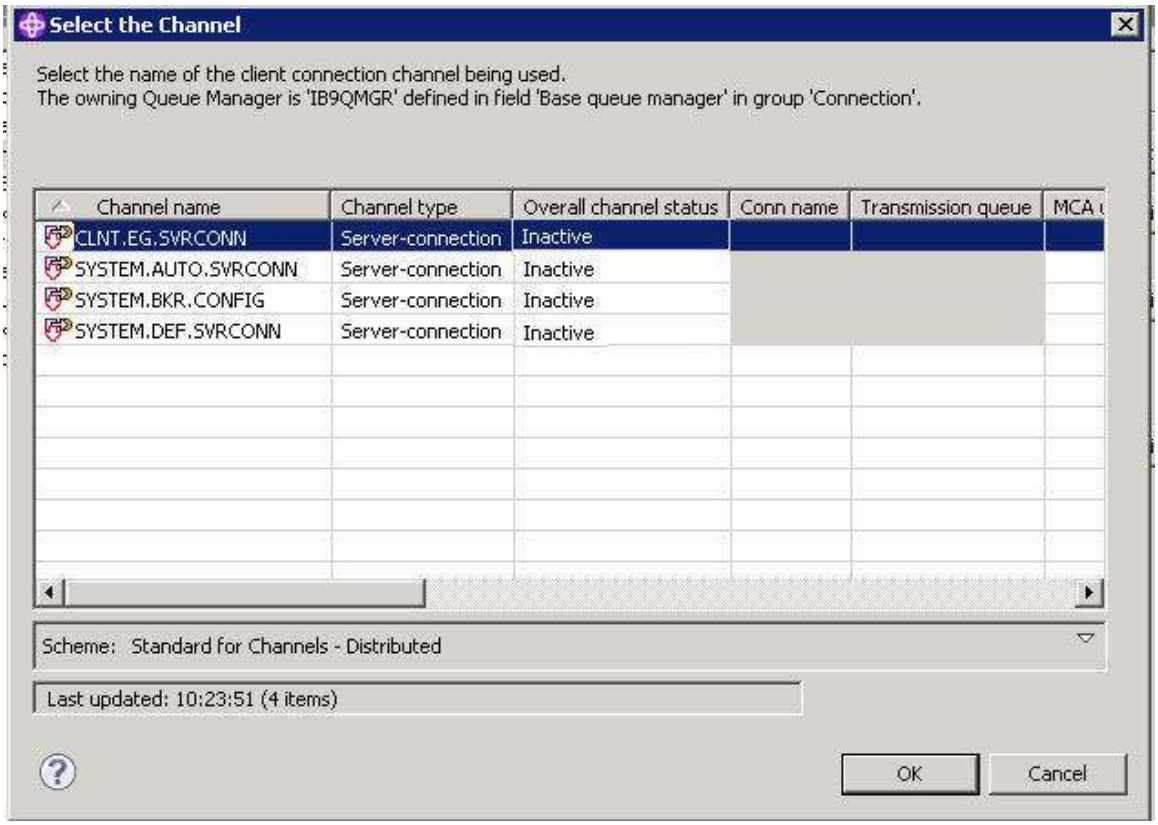

Figure 17: Selecting a server connection channel of your choice

10. Clicking the **OK** button in Figure 17 will populate the **Channel** text box of Figure 16 with the chosen server connection channel as shown in Figure 18.

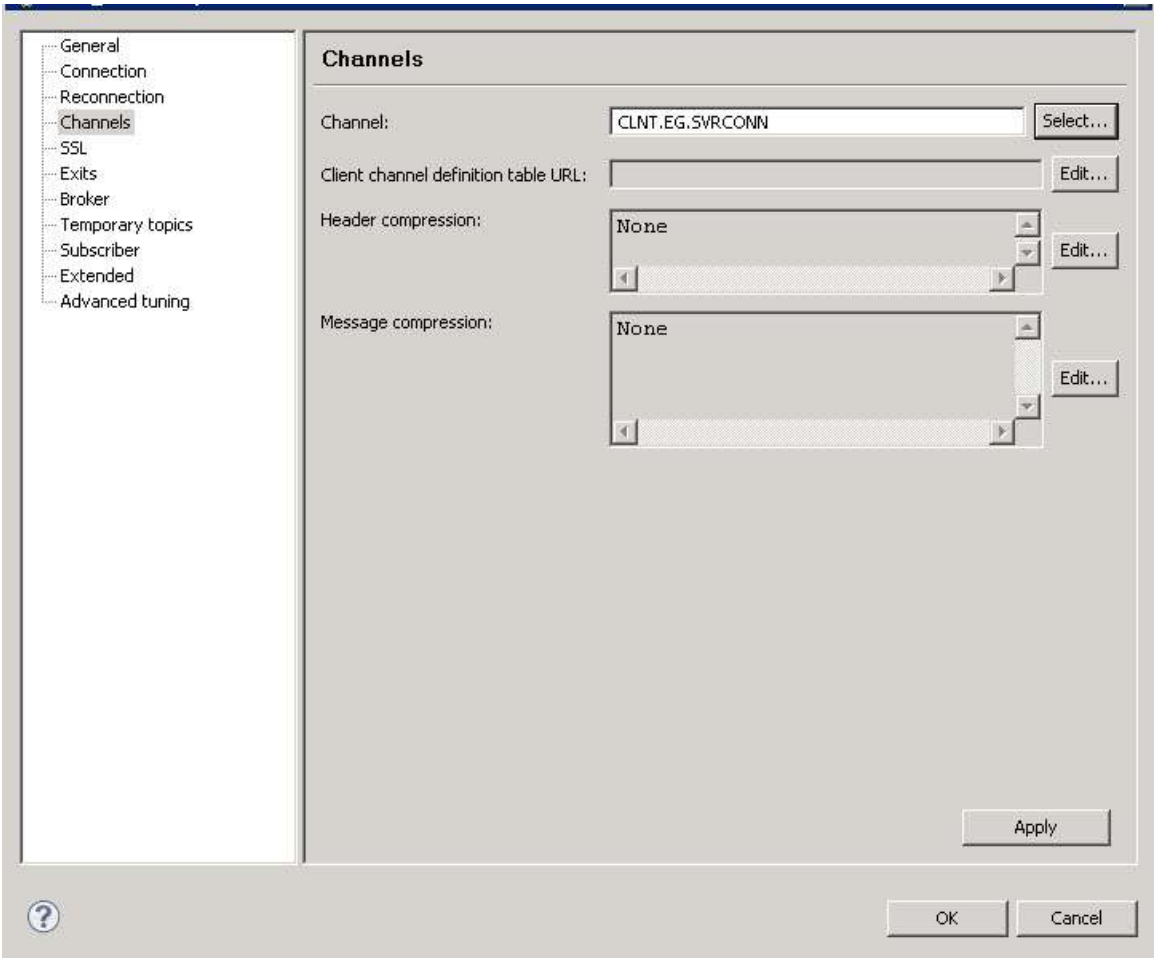

Figure 18: Displaying the chosen server connection channel

11. Clicking the **Apply** button in Figure 18 will let the chosen server connection channel to communicate with the eG agent to collect the required metrics from the target server.

12. To identify the port at which the queue manager is listening, select the **Listeners** node available below the IIB node that is to be monitored in the **MQ Explorer – Navigator** panel (see Figure 19). The port number will be displayed in the **Listeners** panel as shown in Figure 19.

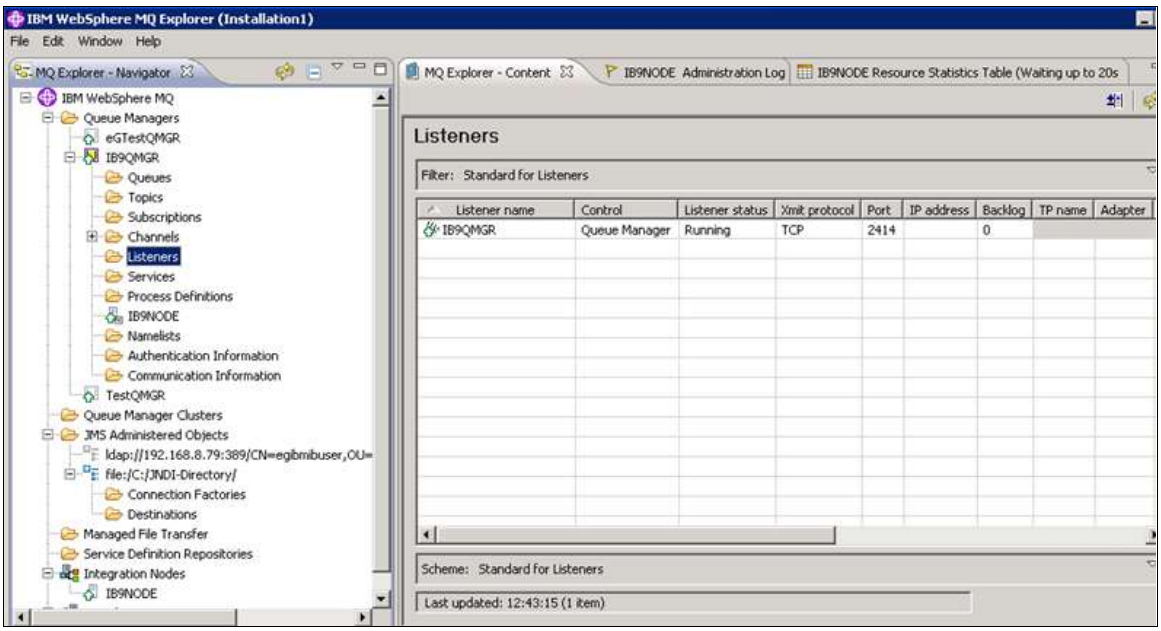

Figure 19: Identifying the port number of the chosen queue manager

13. Clicking the **Finish** button in Figure 15 will create the connection factory successfully.

1.1.3.2.2 Creating the JMS Destinations

By default, the eG agent collects the metrics from the IIB by subscribing to the default topics that are available in the IIB. To store the messages retrieved from the subscribed topics, two new destinations need to be created. Follow the steps below to create the destinations:

1. Right click the **Destinations** folder of Figure 9 and follow the menu sequence: *New -> Destination* (see Figure 20).

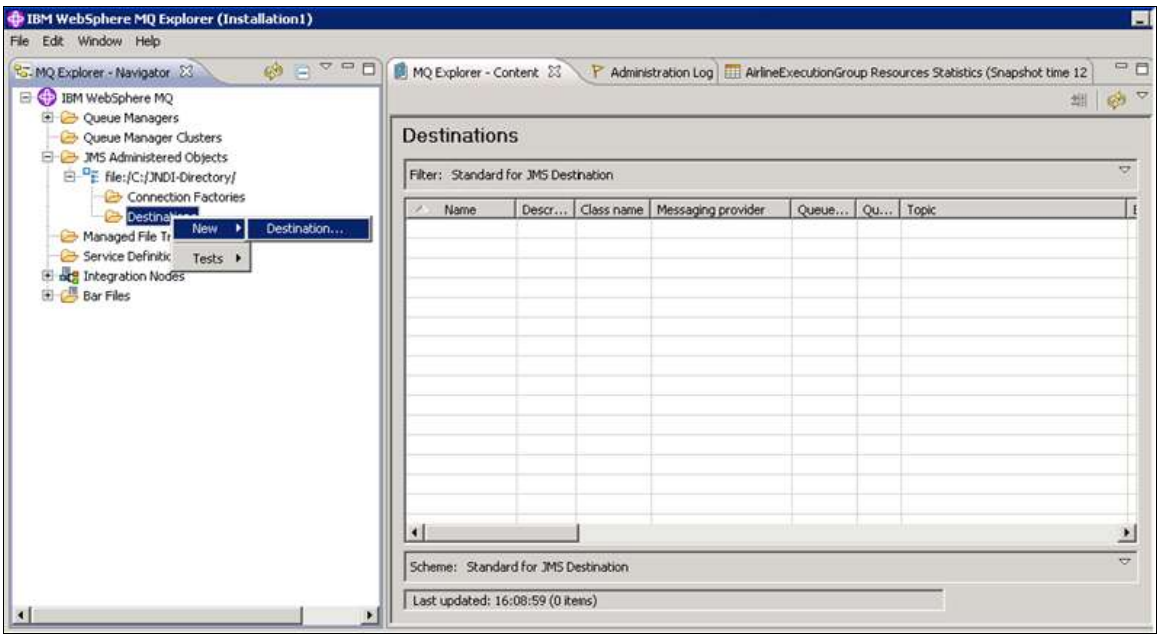

Figure 20: Creating a new JMS destination

2. In Figure 21 that appears, enter the **Name** of the destination that you wish to create. Choose the destination **Type** i.e., the **Topic** that you wish to subscribe.

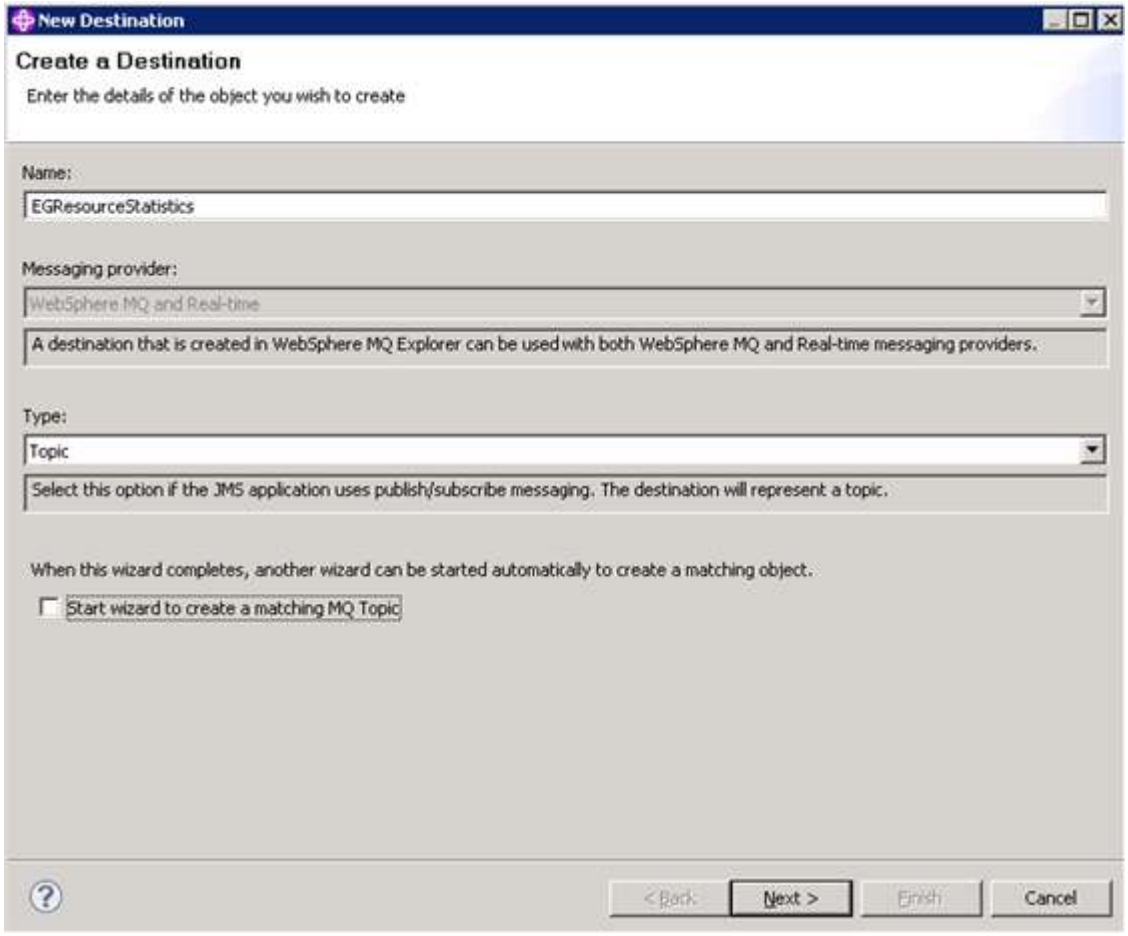

Figure 21: Providing the name of the JMS destination

3. Clicking on the **Next** button in Figure 21 will lead you to the **Change properties** page as shown in Figure 22.

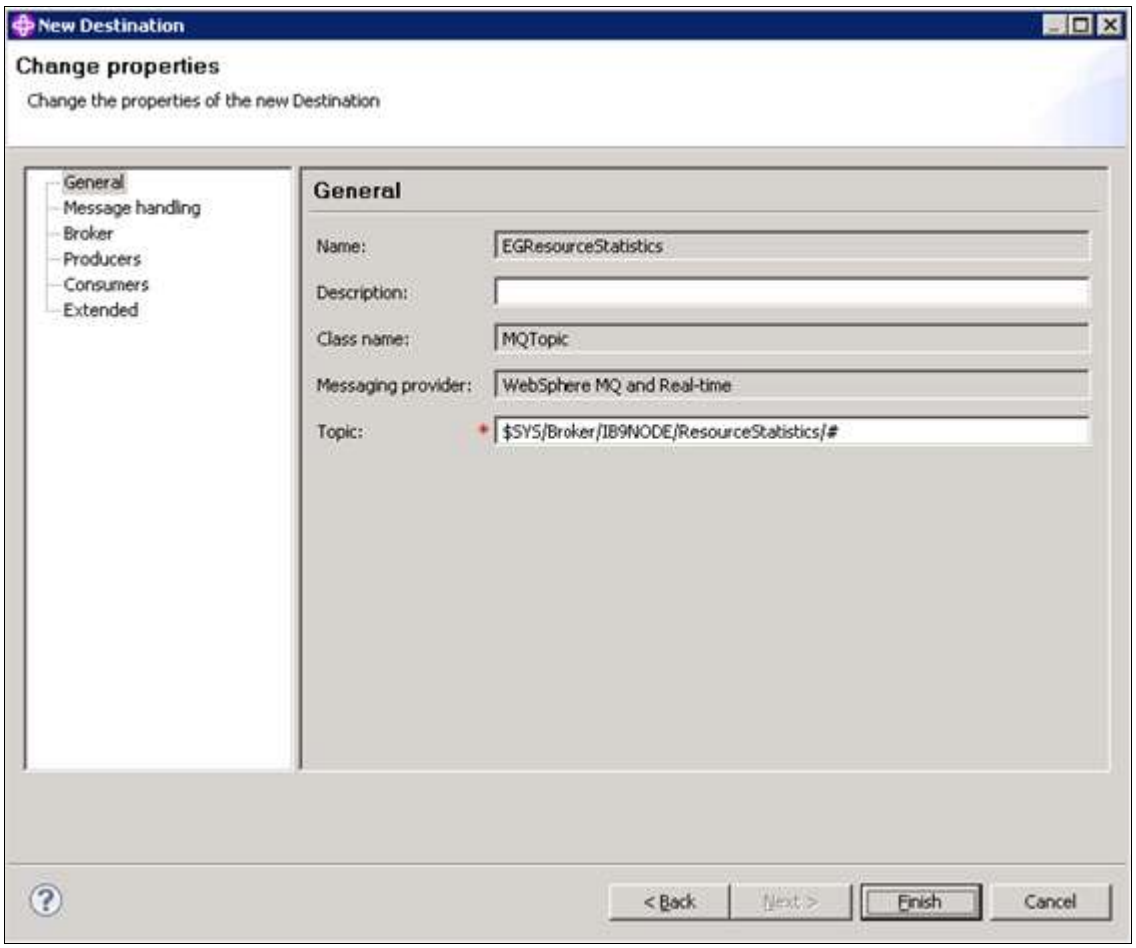

Figure 22: Providing the Topic for storing the resource statistics

4. By default, message flow statistics are stored in one destination and the resource statistics are stored in another destination. To create a destination to store the resource statistics, provide the following string in the **Topic** text box of Figure 22.

```
$SYS/Broker/<Name of the IIB node>/ResourceStatistics/#
```

In our example, the string will be as follows:

```
$SYS/Broker/IB9NODE/ResourceStatistics/#
```

5. Clicking the **Finish** button in Figure 22 will create the desired JMS destination.
6. In order to create a new destination named **EGStatisticsAccounting**, follow the steps 1-3 mentioned above. In the **Change properties** page as shown in Figure 23, provide the following string in the **Topic** text box.

```
$SYS/Broker/<name of the IIB node>/StatisticsAccounting/#
```

In our example, the string will be

```
$SYS/broker/IB9NODE/StatisticsAccounting/#
```

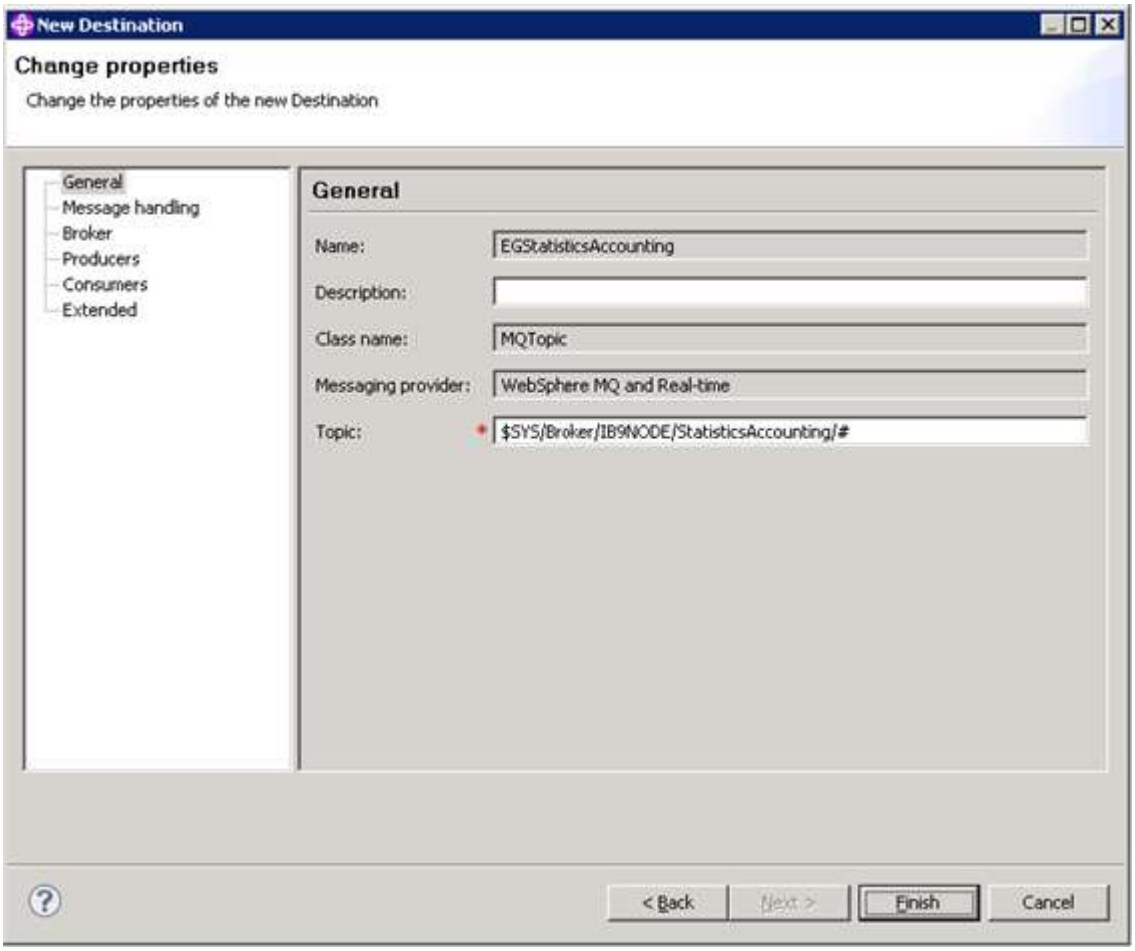

Figure 23: Providing the Topic for collecting the message flow statistics

7. Clicking the **Finish** button in Figure 23 will create the JMS destination successfully.

1.1.4 Agentless Monitoring of the IIB server

If the eG agent is required to monitor the IIB server from a remote location, then the eG agent communicates with the IIB server through an Active Directory server. This way, the security of the eG agent and the IIB server communication is not compromised. Prior to monitoring the IIB server in an agentless manner, you will be required to create a user in the Active Directory server.

1.1.4.1 Creating a user in the Active Directory server

If the eG agent monitors the IIB server from a remote location, then the Active Directory user needs to be created using which the JMS Administered Object is stored. By default, the user attributes store the JMS Administered Object. Follow the below-mentioned steps to create a new user in the Active Directory server prior to creating a JMS Administered Object:

1. Login to the Active Directory server host and follow the menu sequence mentioned in Figure 24.

Figure 24: Following the menu sequence in the Active Directory server host

2. The **Active Directory Users and Computers** page will then open listing the domain on which the Active Directory server operates. In our example, the name of the domain is **DIATEST.COM**. Follow the menu sequence mentioned in Figure 25 to create a new **Organizational Unit** in the domain.

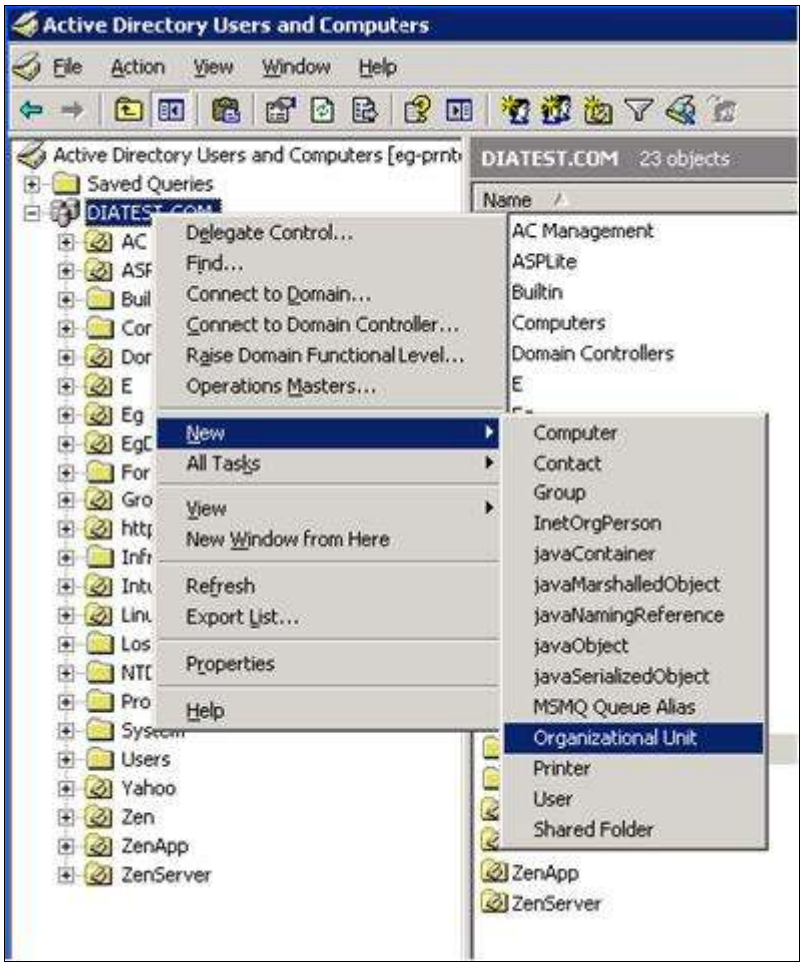

Figure 25: Creating a new Organizational Unit

3. Figure 26 will then appear where you would be required to specify the name of the **Organizational Unit.**

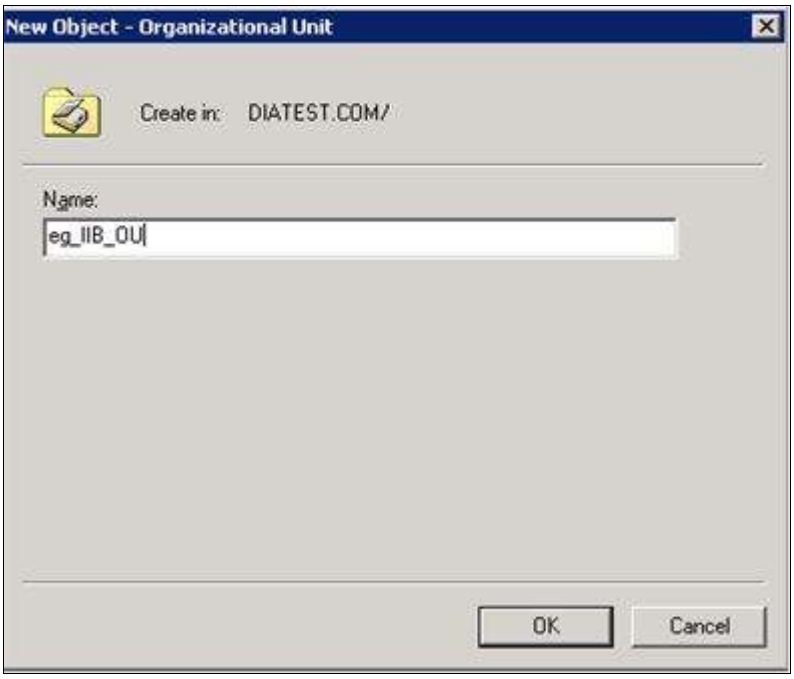

Figure 26: Specifying the name of the Organizational Unit

4. Specify the **Name** of the **Organizational Unit** that you wish to create in Figure 26 and click the **Ok** button. In our example, the **Name** of the **Organizational Unit** is **eg_IIB_OU**.

5. The **Organizational Unit** will now be created and will be listed under the **DIATEST.COM** node as shown in Figure 27. If the eG agent is required to communicate with the IIB server through the Active Directory server, a new user needs to be created whose attributes are authenticated by the IIB server while being monitored by the eG agent. The user attributes store the message details of the connection factory and destinations of the IIB server which are then extracted by the eG agent during monitoring. Follow the menu sequence in Figure 27 to create a new **User**.

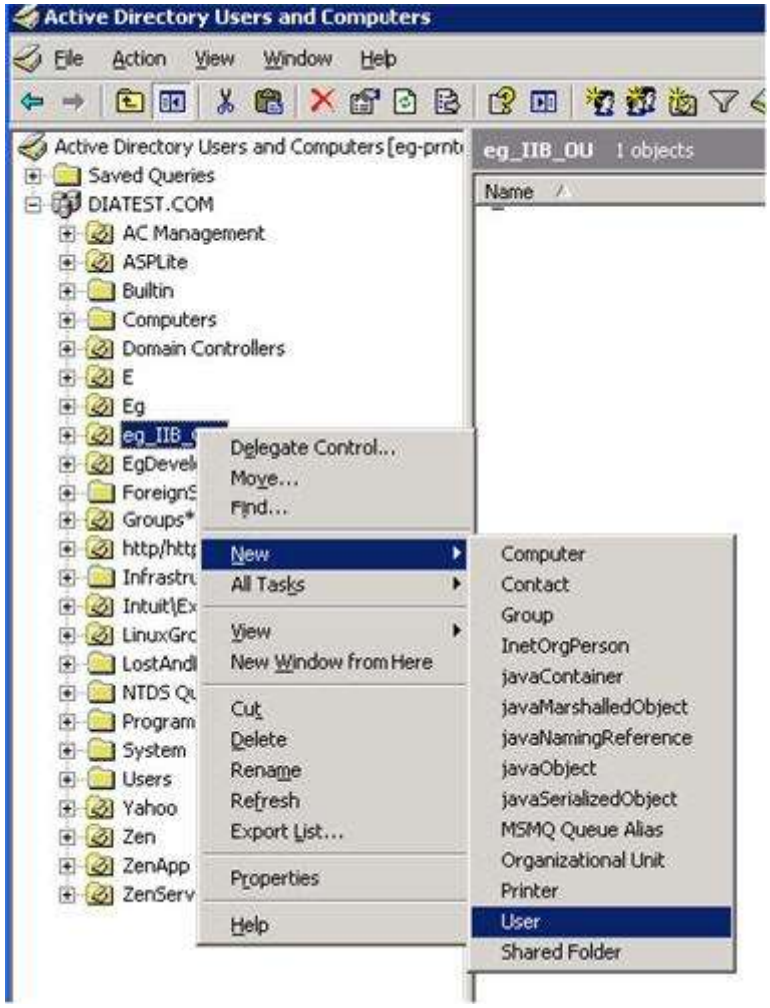

Figure 27: Creating a new user

6. Figure 28 then appears. Specify the **First name** of the user and the **User logon name** of the user in Figure 28. In our example, the **First name** and **User logon name** of the user is **egibmibuser**.

Figure 28: Specifying the name of the user

7. Clicking the **Next** button in Figure 28 will lead you to Figure 29 where you would be required to specify the **Password** for the new user. Confirm the password by retyping it in the **Confirm password** text box and select the **Password never expires** check box.

Figure 29: Providing the password for the new user

8. Figure 30 will then appear listing the credentials of the user .Clicking the **Finish** button in Figure 30 will ensure the successful creation of the user.

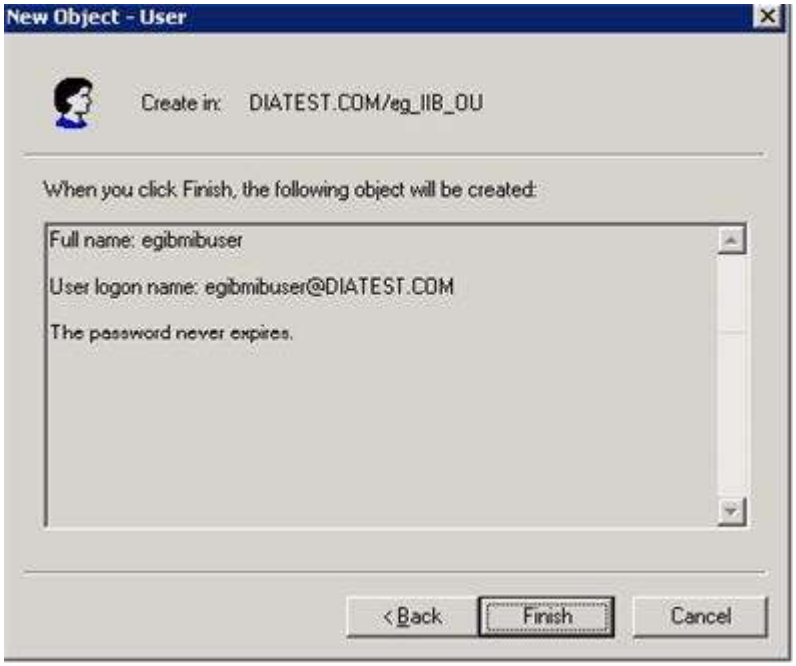

Figure 30: Displaying the credentials of the newly created user

9. Once the user is created, the attributes of the user will store the messages published by the IIB server. The eG agent will communicate with the IIB server through the Active Directory server and collect the required metrics for monitoring from these attributes.

1.1.4.2 Automatically creating a JMS Administered Object using scripts

Follow the steps mentioned below to automatically create a JMS Administered Object:

1. By default, the following files will be available in the **<eG_INSTALL_DIR>\lib** folder say for e.g., **C:\egurkha\lib** of the target environment:
   - eGJMSIIBCreateConfig.bat
   - eGJMSIIBCreateConfig.sh
2. If the IIB server is installed on a Windows environment, execute the **eGJMSIIBCreateConfig.bat** file from the command prompt of the IIB server host as follows:

```
C:\egurkha\lib\>eGJMSIIBCreateConfig.bat
```

   Then proceed to step 5.
3. If the IIB server is installed on a Linux environment, execute the **eGJMSIIBCreateConfig.sh** file from the shell script.

```
/opt/egurkha/lib/>eGJMSIIBCreateConfig.sh
```

4. Now, specify the exact location on which the MQ Java installation directory is installed. In our example, the installation path is **/opt/mqm/java/**.

```
Please enter the MQ java installed path: /opt/mqm/java/
```

5. Once the respective files for the Windows and Linux environments are executed successfully, you will be asked to provide the input parameters that are required for creating the JMS Administered Object. To begin with,

specify the location of the **JNDI namespace** as mentioned below:

```
Where is the JNDI namespace located?
1. File System
2. LDAP Server
Enter your choice [1]: 2
```

6. Selecting the **LDAP Server** option indicates that the **JNDI namespace** will be created on the Active Directory server. Once you have chosen the **LDAP Server** option, you will be required to provide the IP address of the **LDAP server**.

```
Enter the IP address of the LDAP server: 192.168.8.79
```

7. Next, specify the port number of the LDAP server.

```
Enter the LDAP Listener port: 389
```

8. Then, specify whether the LDAP server is SSL enabled. By default, the LDAP server is not SSL enabled.

```
Is the LDAP server SSL enabled? Yes/No [No]: No
```

9. Next, specify the credentials of the user that you have created in Section 1.1. The credentials of the user should be in the format: **cn=<name of the user>,ou=<name of the Organizational Unit>,dc=<name of the domain controller>**. In our example, the user credentials will be as follows:

```
Enter the User DN: cn=egibmibuser,ou=eg_IIB_OU,dc=DIATEST,dc=COM
```

10. Enter the password of the user that you have specified while creating the user (see Figure 6).

```
Enter your password:**********
```

11. Once all the inputs for the LDAP server are provided, you will be specified to provide the IP address of the IIB server host.

```
Enter the host IP address on which the IBM Integration Bus is installed:
192.168.9.165
```

12. To create the **Connection Factories** and the **Destinations**, specify the name of the queue manager.

```
Enter the name of the QManager: IB9QMGR
```

13. Now, specify the name of the node in the IIB server host that is to be monitored as shown below:

```
Enter the name of the Integration node: IB9NODE
```

14. Finally, specifying the port on which the **QManager** listens will automatically create the **JMS Administered Object** along with the **Connection Factories** and the **Destinations**.

```
Enter the port on which the QManager listens: 2414
```

15. If all the input parameters are successfully validated, then the following details will appear.

```
********************************************************************************
Following are the parameters that need to be provided while configuring the tests:
```

```
HOST = 192.168.9.165
QMANAGER NAME = IB9QMGR
QMANAGER PORT = 2414
INTEGRATION NODE NAME = IB9NODE
JNDI NAMESPACE LOCATION = LDAP
SSL = No
USERDN ==> cn=egibmibuser,ou=eg_IIB_OU,dc=DIATEST,dc=COM
JNDI PROVIDER URL ==>
ldap://192.168.8.79:389/cn=egibmibuser,ou=eg_IIB_OU,dc=DIATEST,dc=COM
TOPICCONNECTIONFACTORY = eG_TCF_192.168.9.165_2414
JMS RESOURCE STATS = eG_Res_Topic_192.168.9.165_2414
JMS FLOW STATS = eG_Msg_Topic_192.168.9.165_2414
************************************************************************
```

16. Specify the values mentioned above against the appropriate fields while configuring the tests for the IIB server that is to be monitored. To know how to configure the tests, refer to Chapter 2 of this document.

If you fail to notice the details mentioned in Step 15, then you can figure them out in the **<EG_INSTALL_DIR>\lib\IIB\eG_IIB_test_config.log** file. If multiple IIB nodes are monitored in the target environment, then the test parameter values will be stored for each IIB node separately.

1.1.4.3 Manually creating a JMS Administered Object in the IIB server using MQ Explorer console/GUI

Follow the steps mentioned below to create a JMS Administered Object:

1. From the IIB server host, follow the menu sequence: *Start -> All Programs -> IBM Integration Bus 9.0 -> IBM Integration Explorer* (see Figure 31).

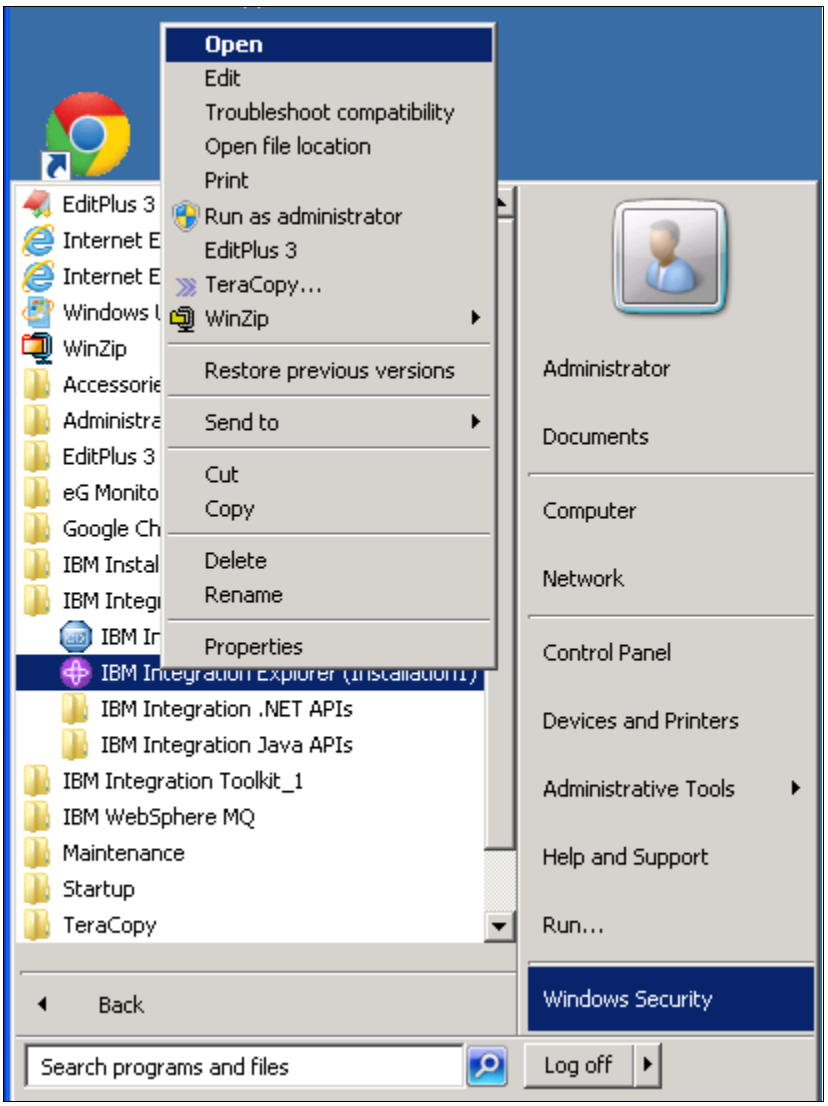

Figure 31: Opening the IBM Integration Explorer menu

2. Open the **IBM Integration Explorer** as shown in Figure 31. Figure 32 will then appear.

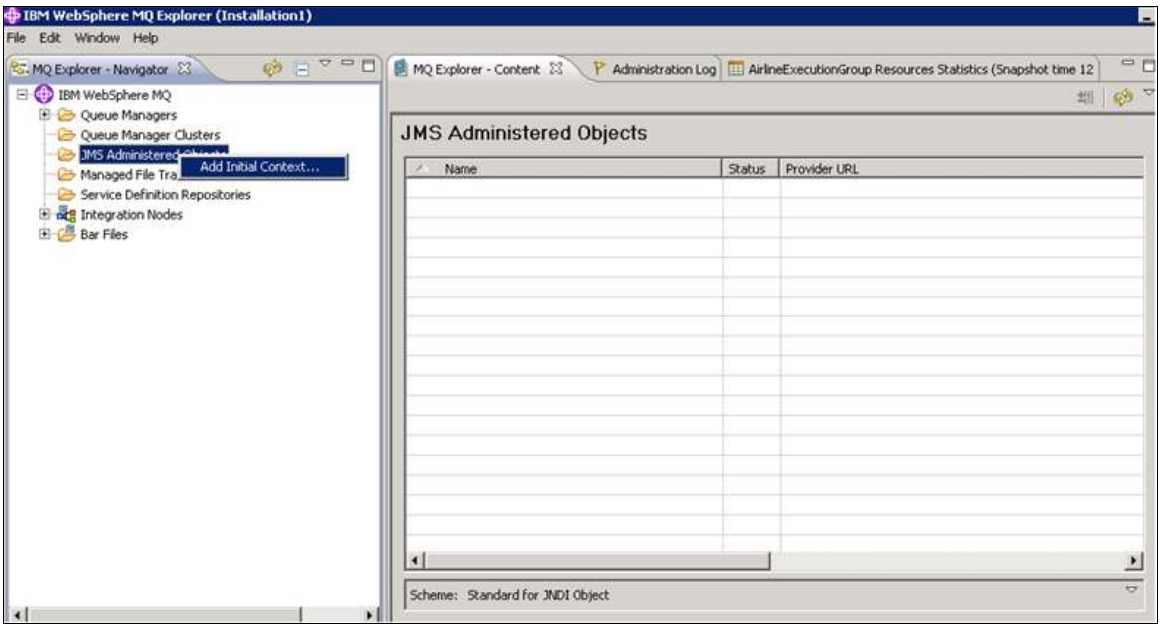

Figure 32: Opening the IBM WebSphere MQ Explorer

3. Right clicking the **JMS Administered Objects** node from the **MQ Explorer – Navigator** panel will display the **Add Initial Context** option (see Figure 32). An initial context is the root of the JNDI namespace that is used to access the JMS objects that are stored in them.

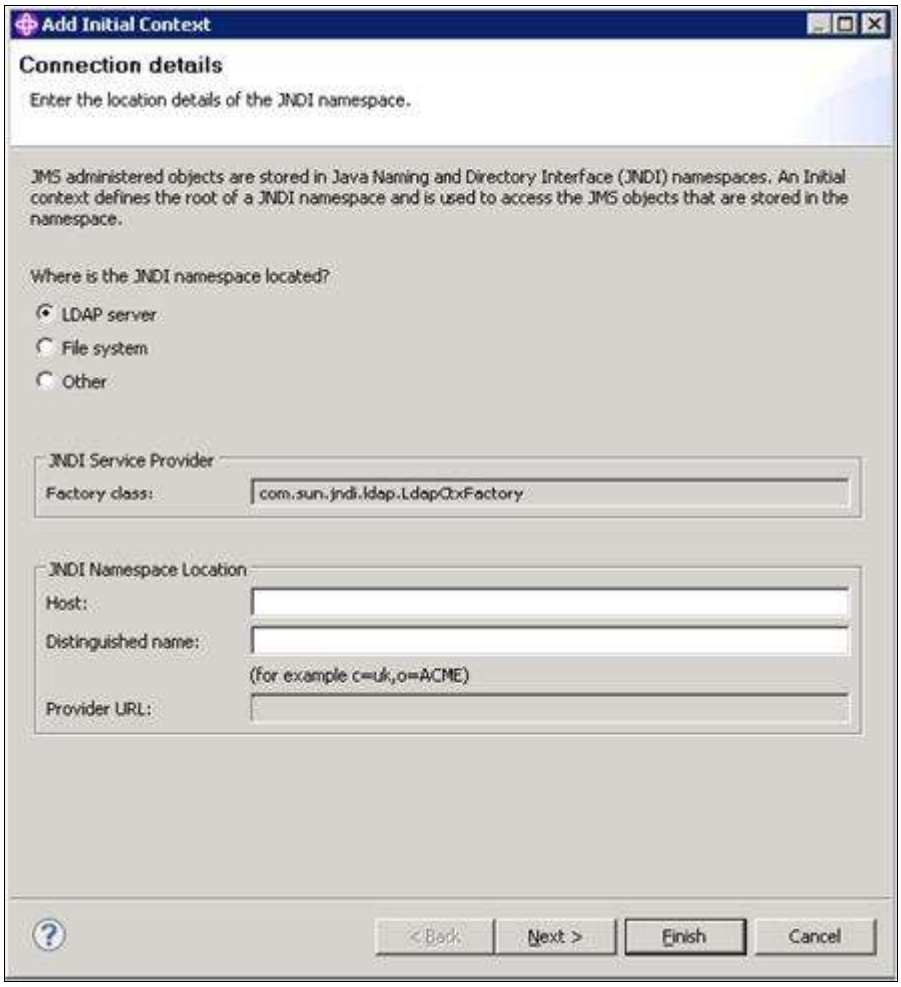

Figure 33: Choosing the location of the JNDI namespace

4. In Figure 33 that appears, you will be required to provide the location of the JNDI namespace on which the JMS Administered Object is stored. If the eG agent monitors the IIB server from a remote location, then set the **Where is the JNDI namespace located?** to **LDAP server**.

5. Specify the IP address and the Port number of the Active directory server that is used by the eG agent to communicate with the IIB server in the **Host** text box of Figure 33. In the **Distinguished name** text box, specify the credentials of the user that you have created in Section 1.1.4.1. The credentials of the user should be in the format: **CN=<name of the user>,OU=<name of the Organizational Unit>,DC=<name of the domain controller>**. In our example, the user credentials will be as follows: **CN=egibmibuser,OU=eg_IIB_OU,DC=DIATEST,DC=COM**. Once the **Host** and the **Distinguished name** are specified, the **Provider URL** will be displayed automatically as shown in Figure 34.

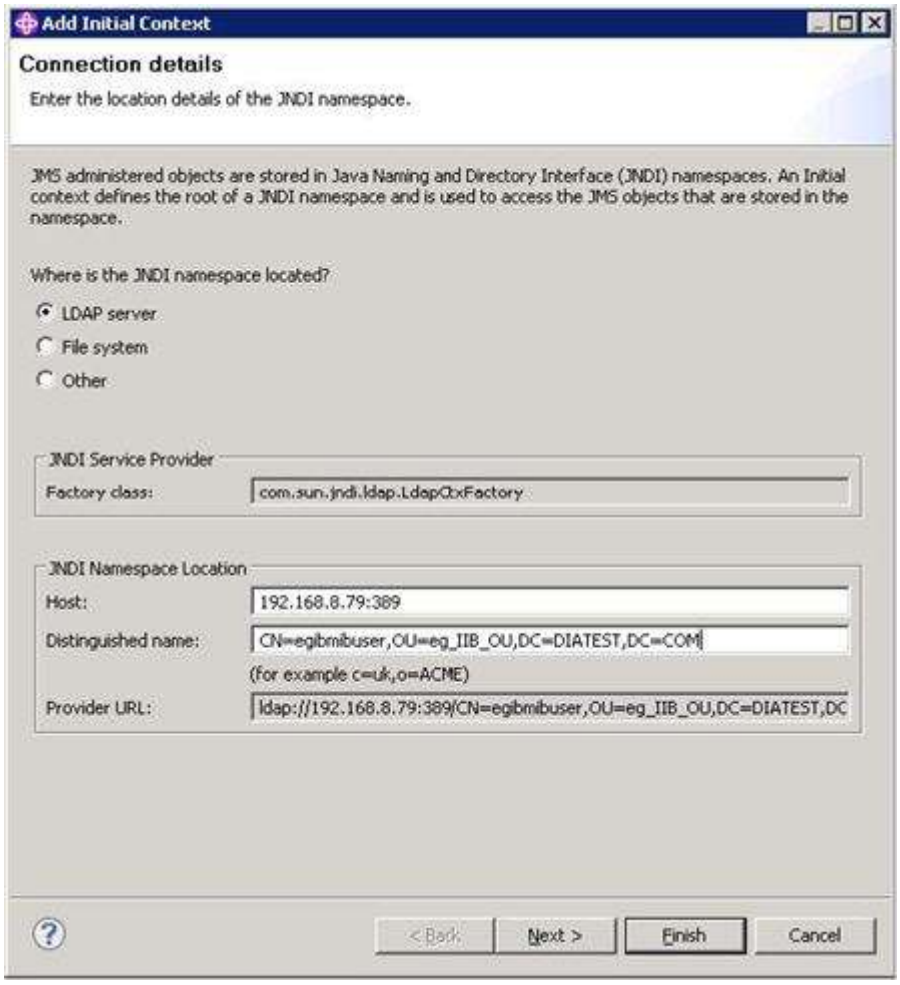

Figure 34: Specifying the exact path of the JNDI namespace

6. Clicking the **Next** button in Figure 34 will navigate you to Figure 35. Provide a name of your choice in the **Context nickname** textbox.

Figure 35: Providing a nickname for the JMS Administered Object to be created

7. Clicking the **Finish** button in Figure 35 will invoke Figure 36. Specify the **User DN** as mentioned in the **Distinguished name** text box of Figure 35 and the **Password** as mentioned in Figure 29. Clicking the **OK** button in Figure 36 will complete the initial context specification.

Figure 36: Specifying the login credentials of the new user

8. The nickname specified in Figure 35 will be displayed as the name of the **JMS Administered object** in the **MQ Explorer – Navigator** section. In our example, the nickname is not specified and the **Provider URL** is automatically displayed as the **JMS Administered Object** with two default folders named **Connection Factories** and **Destinations** (see Figure 36).

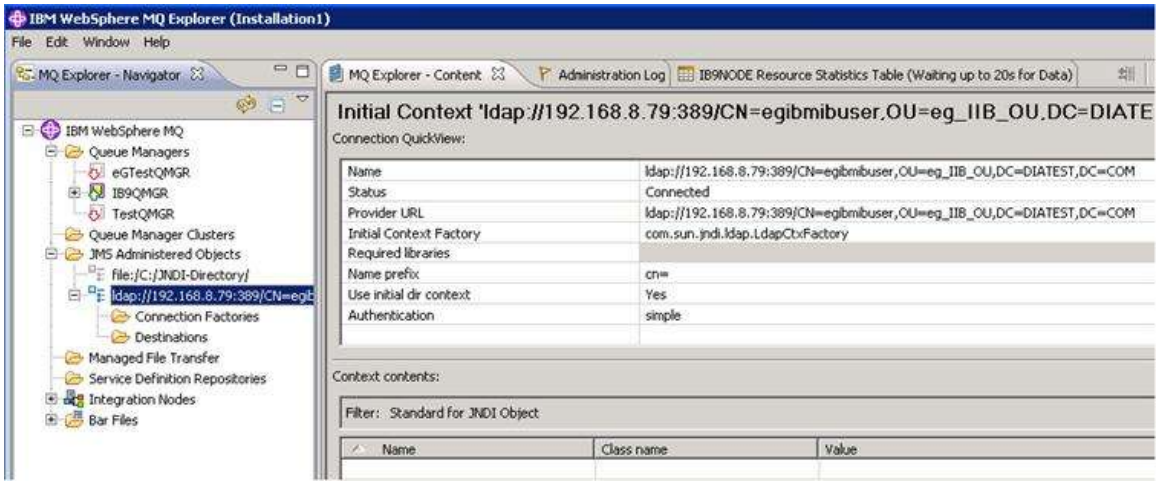

Figure 37: Viewing the created JMS Administered Object

Now let us discuss the steps that need to be followed for creating a connection factory and the destinations.

1.1.4.3.1 Creating a Connection Factory

For an eG agent to connect to the IIB and collect the desired metrics, a connection needs to be established between the eG agent and the Queue manager of the IIB that is to be monitored. To establish such a connection, you would be required to create a connection factory. Follow the below mentioned steps to create a connection factory:

1. Right click **the Connection Factories** folder of Figure 38 and follow the menu sequence: *New -> Connection Factory* (see Figure 38).

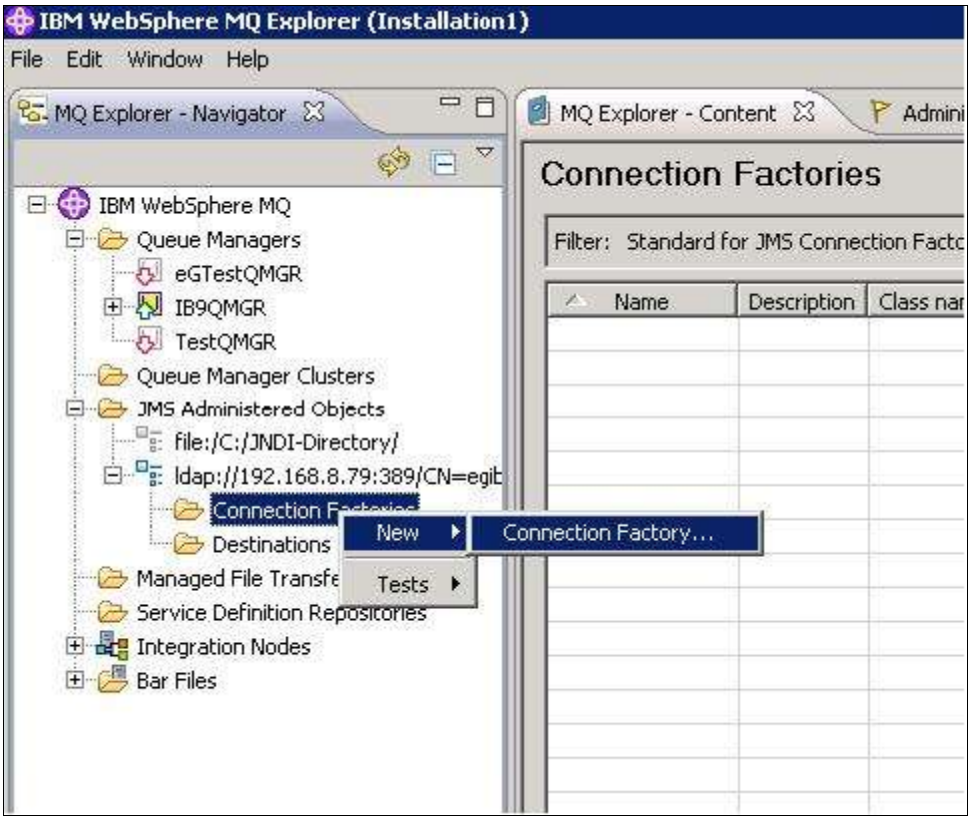

Figure 38: Creating a new connection factory

2. In Figure 39 that appears, specify the **Name** of the connection factory. In our example, the name of the connection factory is **EgTopicConnectionFactory**.

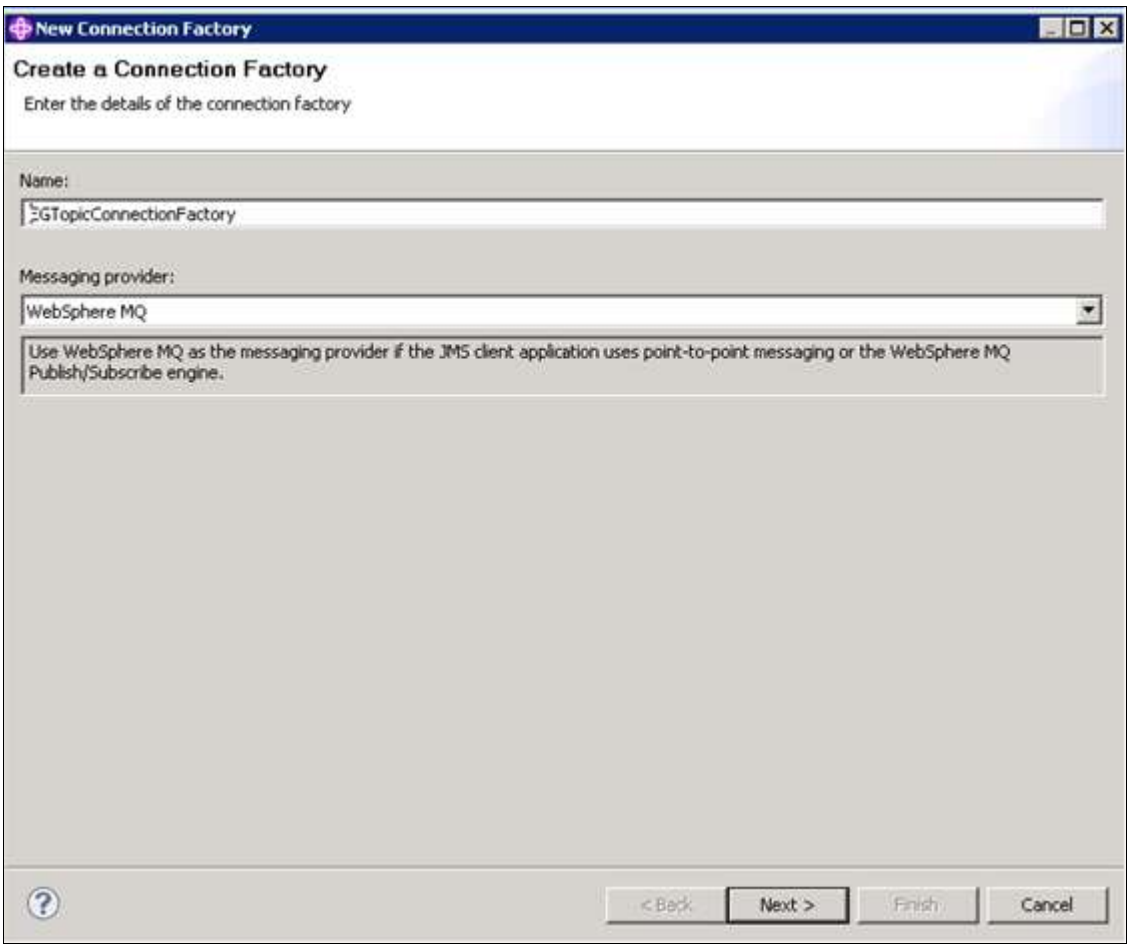

Figure 39: Specifying the name of the connection factory

3. By default, the messages from the IIB are published on the WebSphere MQ. The eG agent is required to subscribe to the messages in order to monitor the IIB. Select **WebSphere MQ** as the **Messaging provider** so that the eG agent can communicate with the WebSphere MQ to collect the required metrics while monitoring the IIB.

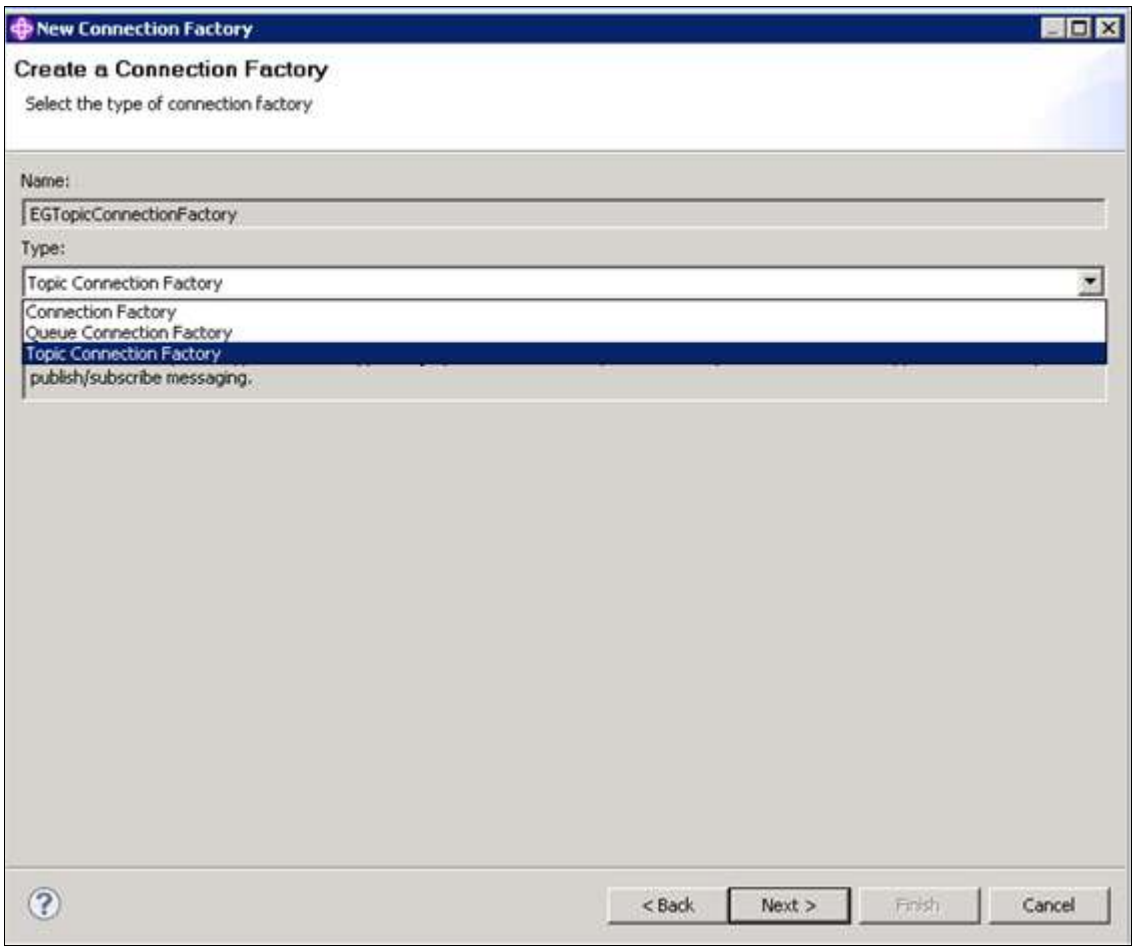

Figure 40: Selecting the type of the connection factory

4. Clicking the **Next** button in Figure 39 will enable you to select the **Type** of the connection factory. Since the eG agent subscribes to the published messages to collect the metrics, select **Topic Connection Factory** option as the **Type**. Clicking the **Next** button will navigate you to Figure 41.

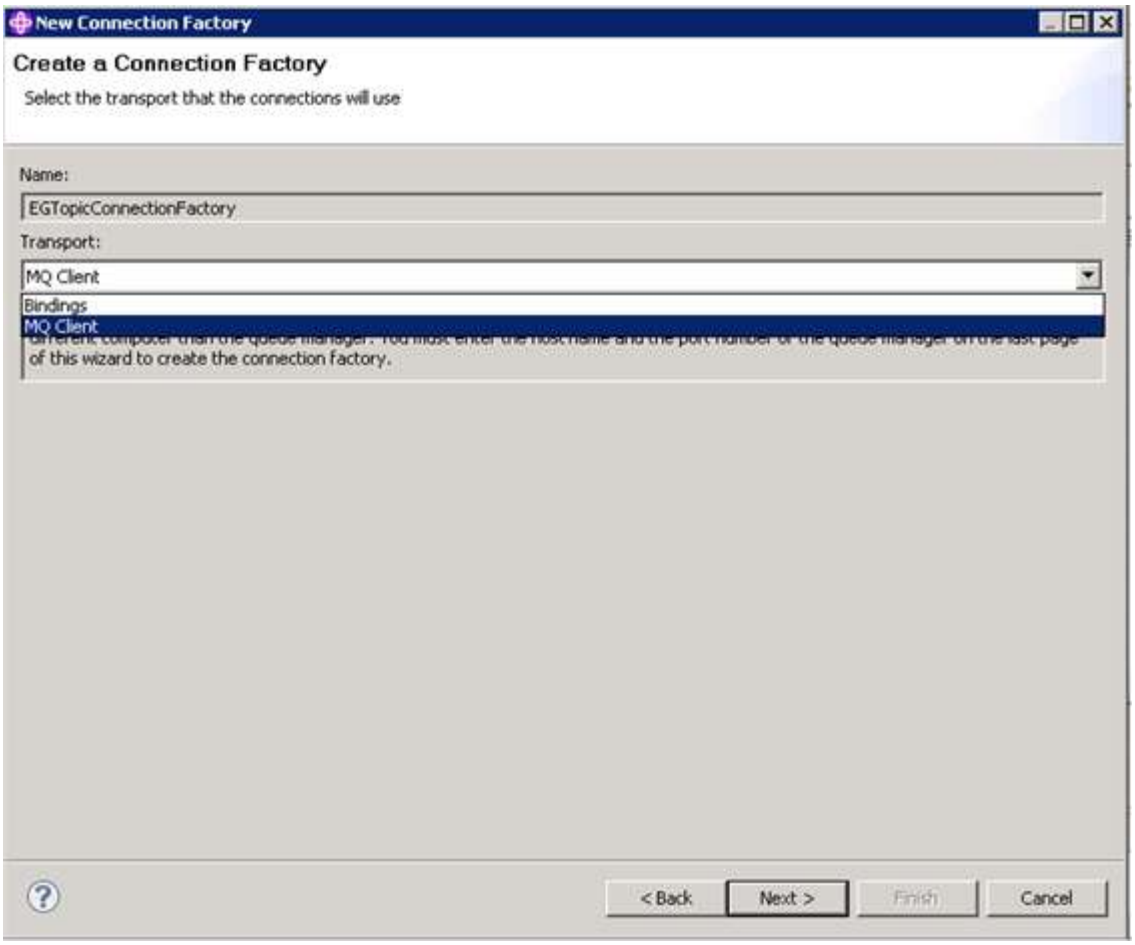

Figure 41: Selecting the transport type of the connection factory

5. Select MQ client as the **Transport** type from Figure 41 and click the **Next** button.

6. The **Change Properties** page will then appear displaying the general settings of the connection factory. Click the **Connection** option in the left panel to select the base queue manager of the IIB node that is to be monitored (see Figure 42). Clicking the **Select**... button will list out all the available Queue managers in the **Select the Base Queue manager** pop up window. Select the queue manager of your choice and click the **OK** button in Figure 42.

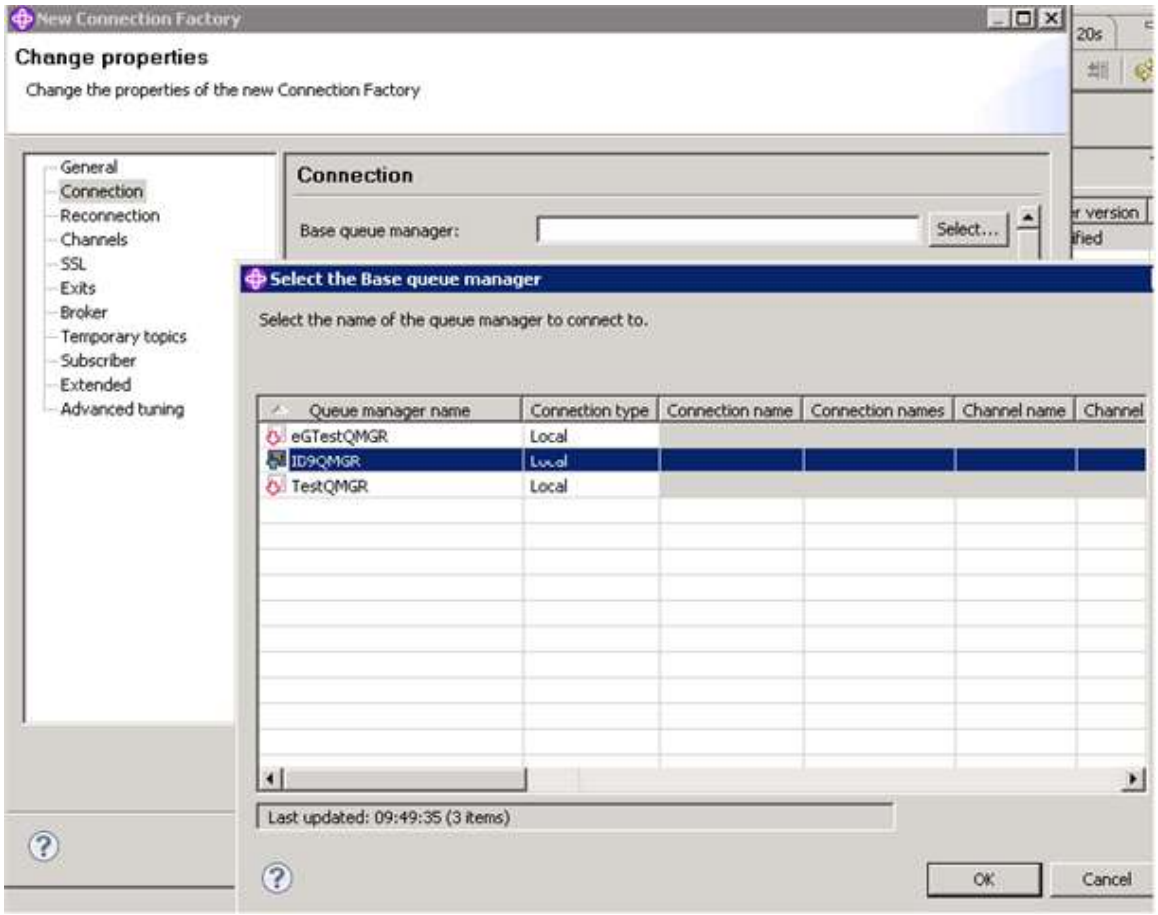

Figure 42: Selecting the Base queue manager

7. In the **Connection list** of Figure 43, specify the host name i.e., the IP address of the chosen queue manager and the port on which the queue manager listens in the following format: **IP address(Port)**.

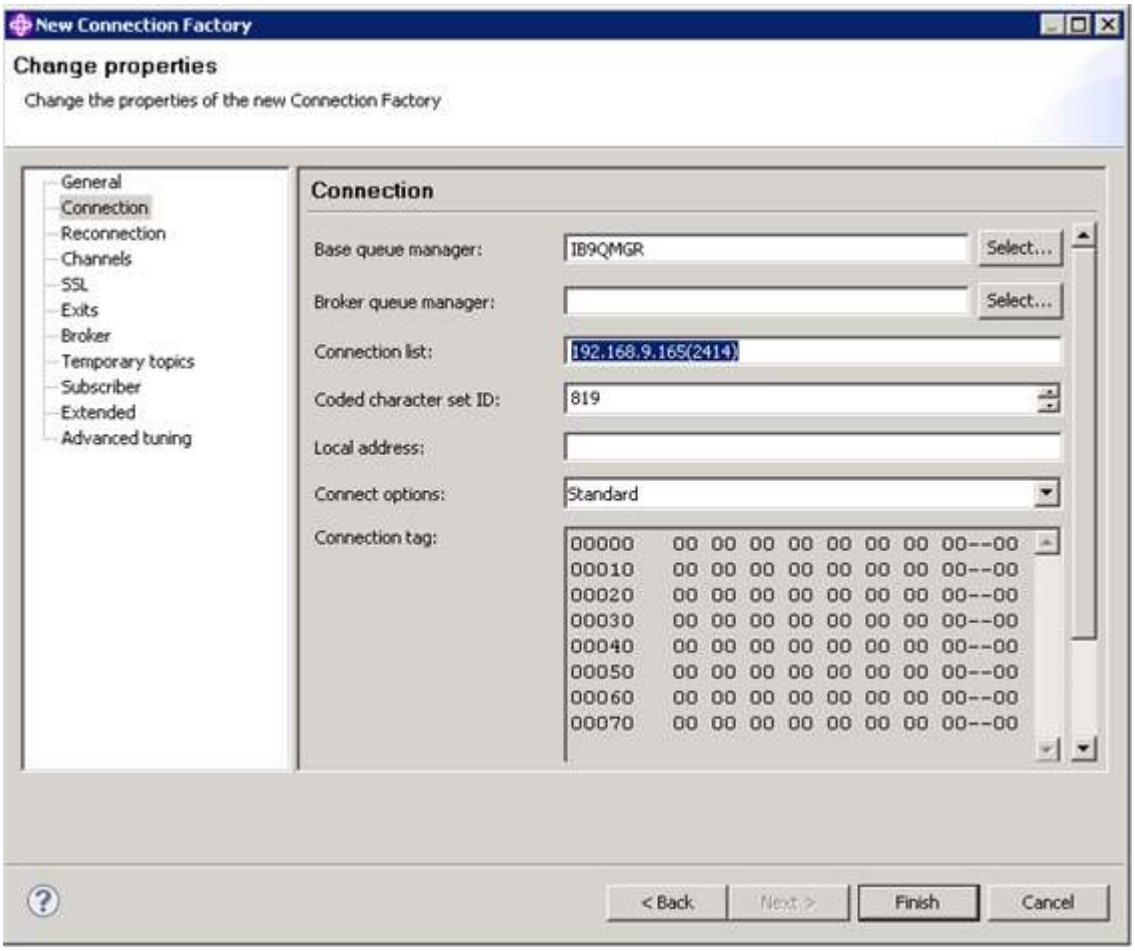

Figure 43: Specifying the host name and port number of the chosen queue manager

8. If you have chosen to use a custom server connection channel for monitoring by the eG Enterprise Suite, then you may be required to select the server connection channel of your choice from the **Channels** option of the **Change Properties** page. The default server connection channel will be listed in the **Channel** text box as shown in Figure 44.

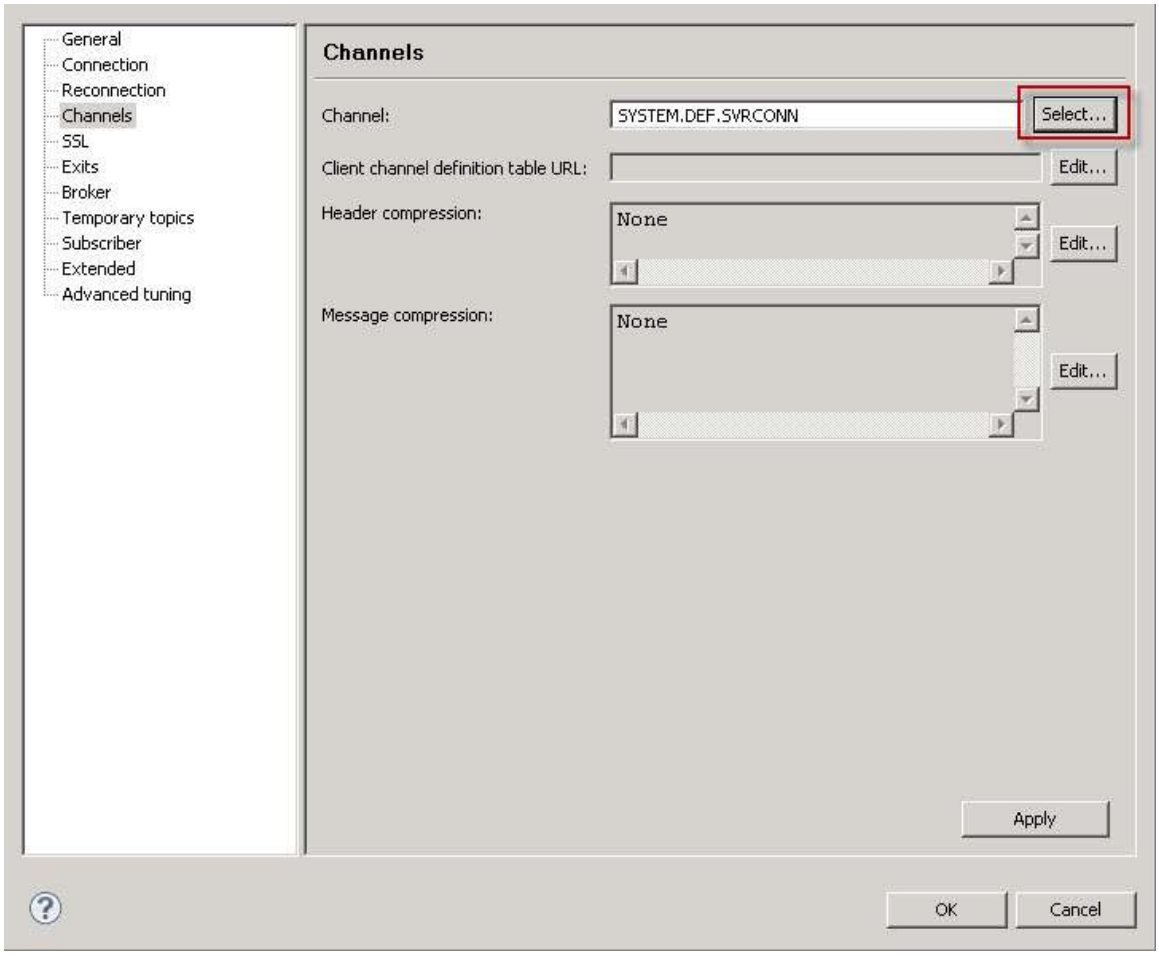

Figure 44: The default server connection channel

9. Clicking the **Select** button against the **Channel** text box in Figure 44 will lead you to Figure 45 where you will be required to choose the server connection channel that you have created for monitoring purpose by the eG Enterprise Suite.

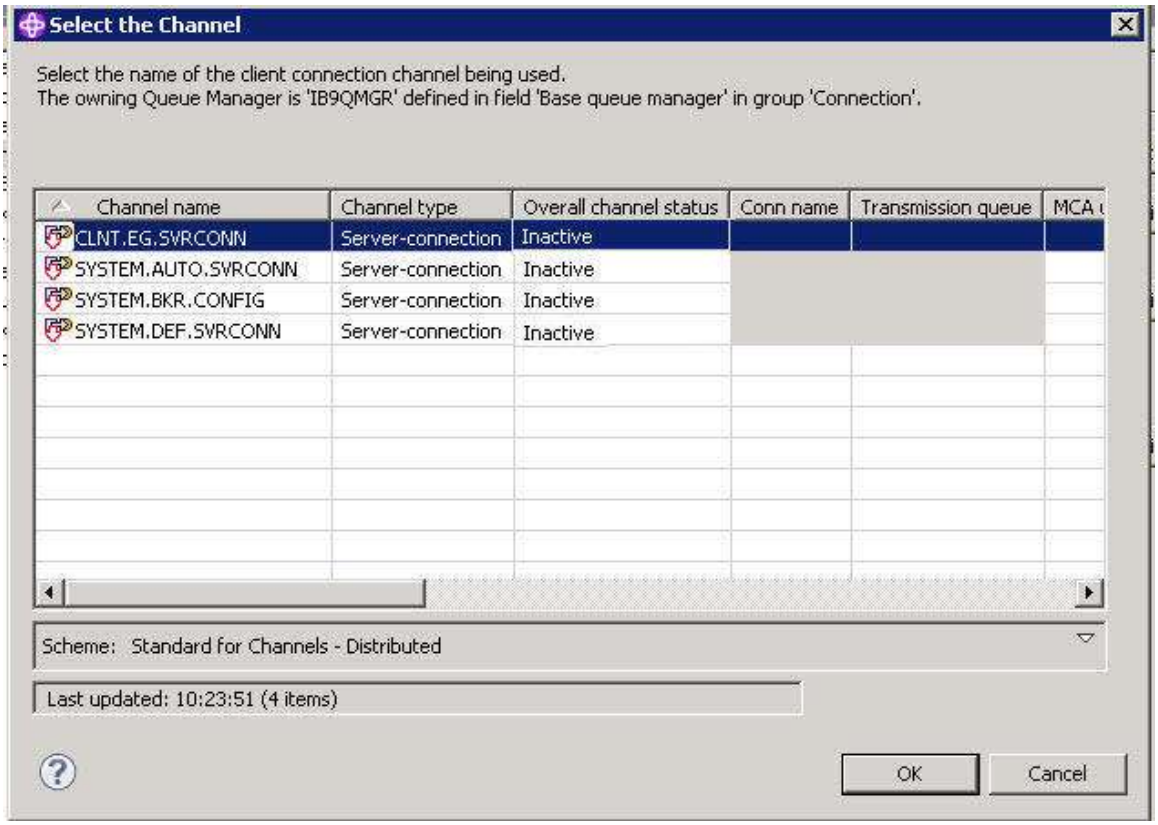

Figure 45: Selecting a server connection channel of your choice

10. Clicking the **OK** button in Figure 45 will populate the **Channel** text box of Figure 44 with the chosen server connection channel as shown in Figure 46.

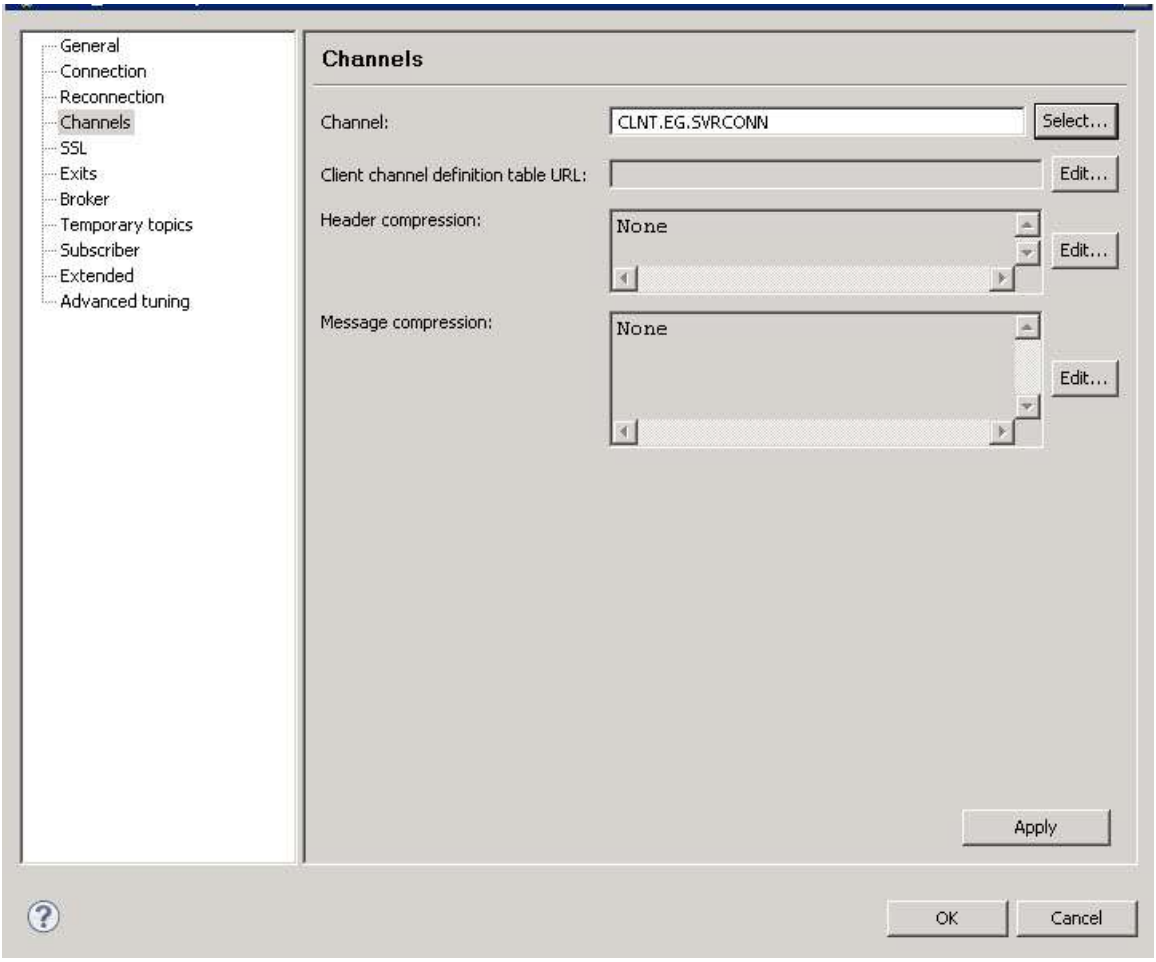

Figure 46: Displaying the chosen server connection channel

11. Clicking the **Apply** button in Figure 46 will let the chosen server connection channel to communicate with the eG agent to collect the required metrics from the target server.

12. To identify the port at which the queue manager is listening, select the **Listeners** node available below the IIB node that is to be monitored in the **MQ Explorer – Navigator** panel (see Figure 47). The port number will be displayed in the **Listeners** panel as shown in Figure 47.

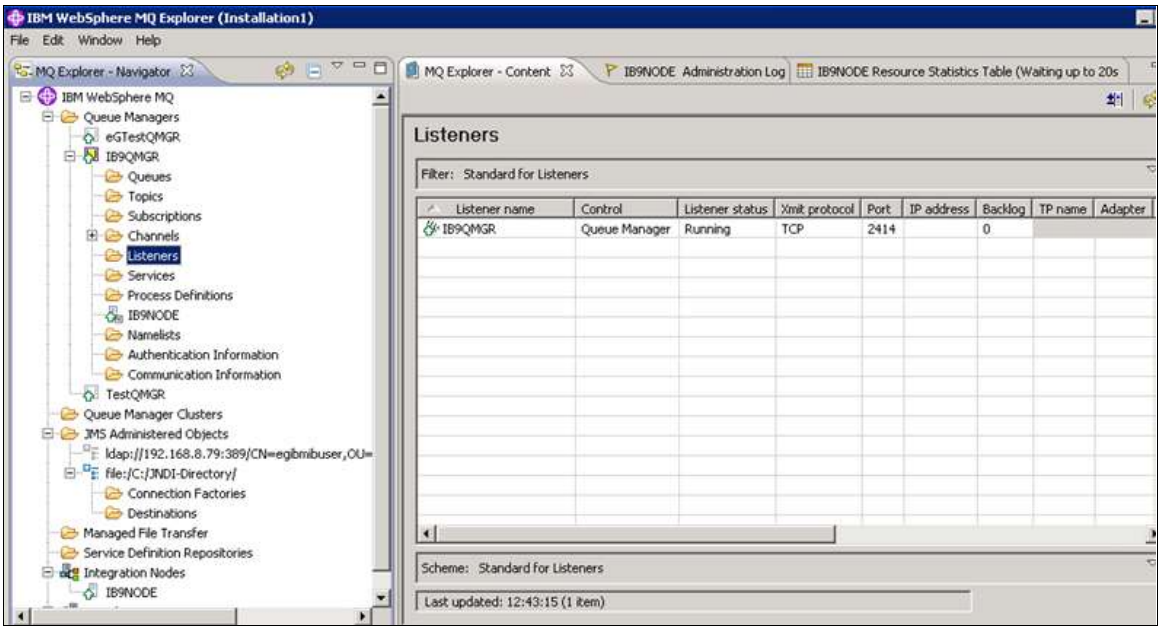

Figure 47: Identifying the port number of the chosen queue manager

13. Clicking the **Finish** button in Figure 43 will create the connection factory successfully.

1.1.4.3.2 Creating the JMS Destinations

By default, the eG agent collects the metrics from the IIB by subscribing to the default topics that are available in the IIB. To store the messages retrieved from the subscribed topics, two new destinations need to be created. Follow the steps below to create the destinations:

1. Right click the **Destinations** folder of Figure 31 and follow the menu sequence: *New -> Destination* (see Figure 48).

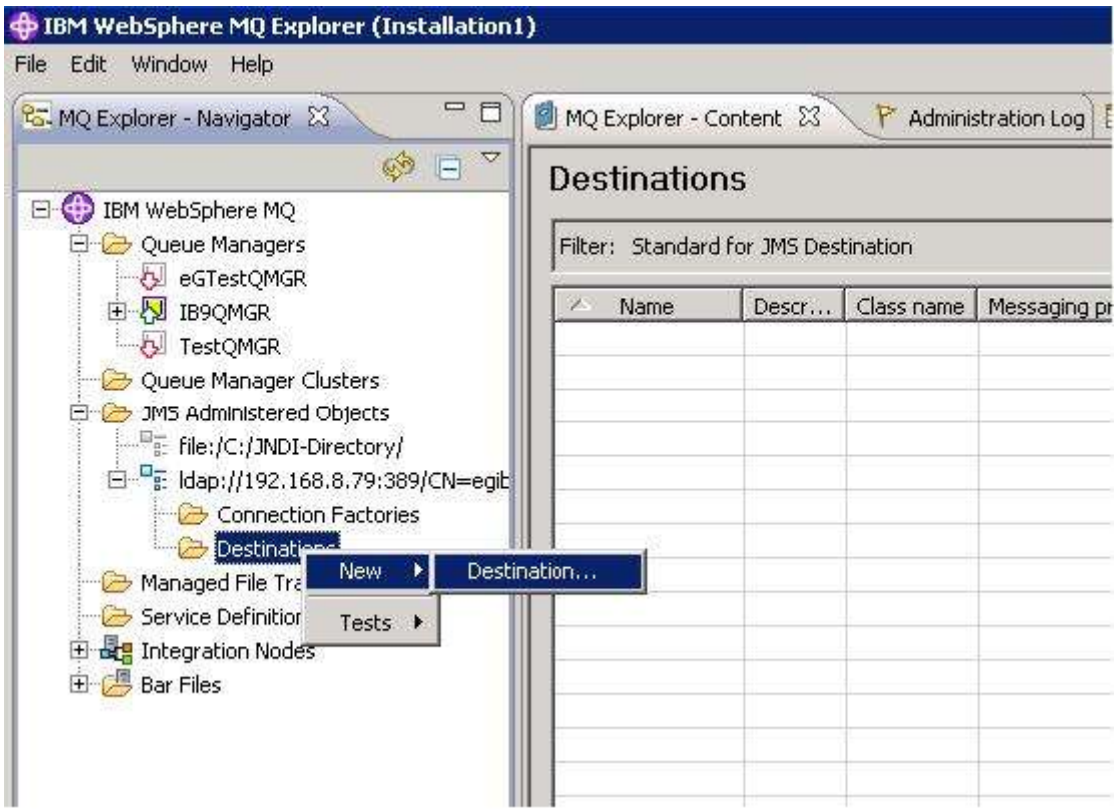

Figure 48: Creating a new JMS destination

2. In Figure 49 that appears, enter the **Name** of the destination that you wish to create. Choose the destination **Type** i.e., the **Topic** that you wish to subscribe.

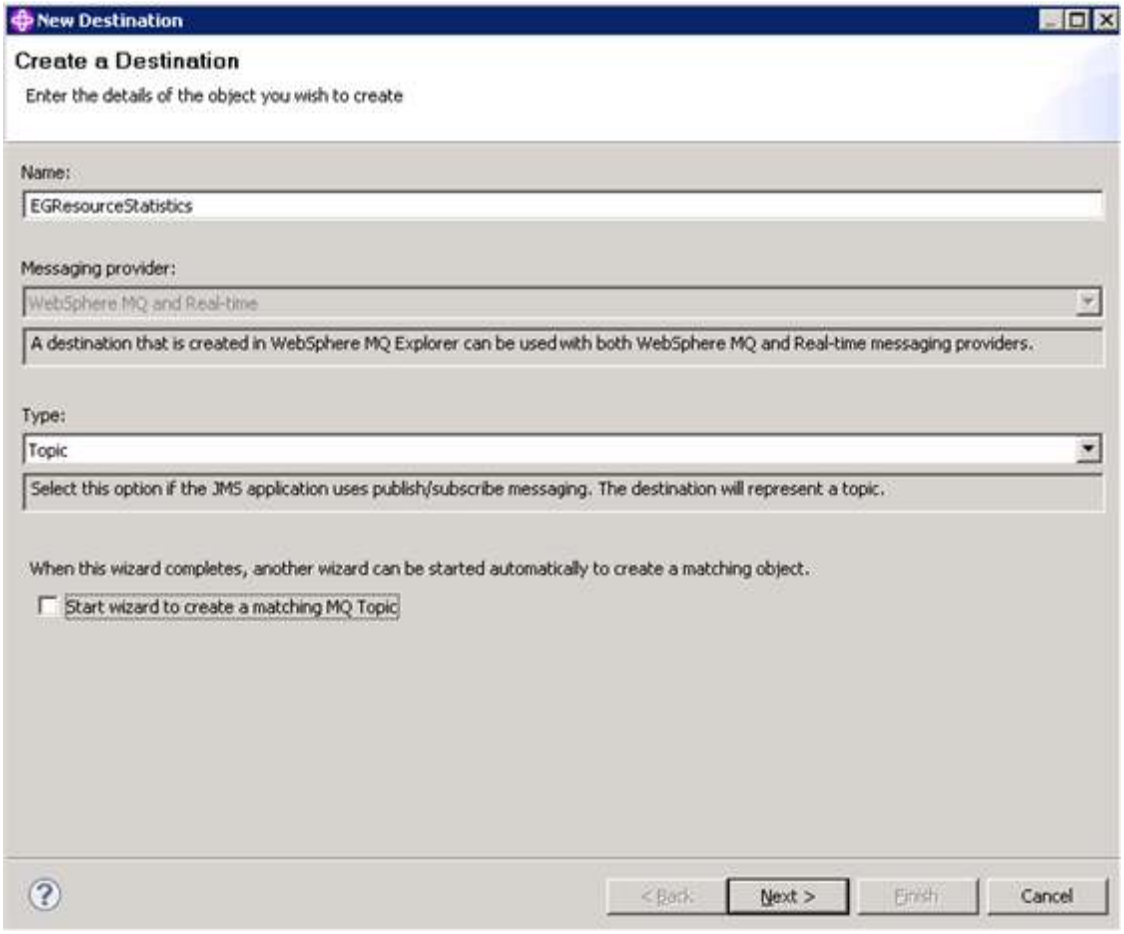

Figure 49: Providing the name of the JMS destination

3. Clicking on the **Next** button in Figure 49 will lead you to the **Change properties** page as shown in Figure 50.

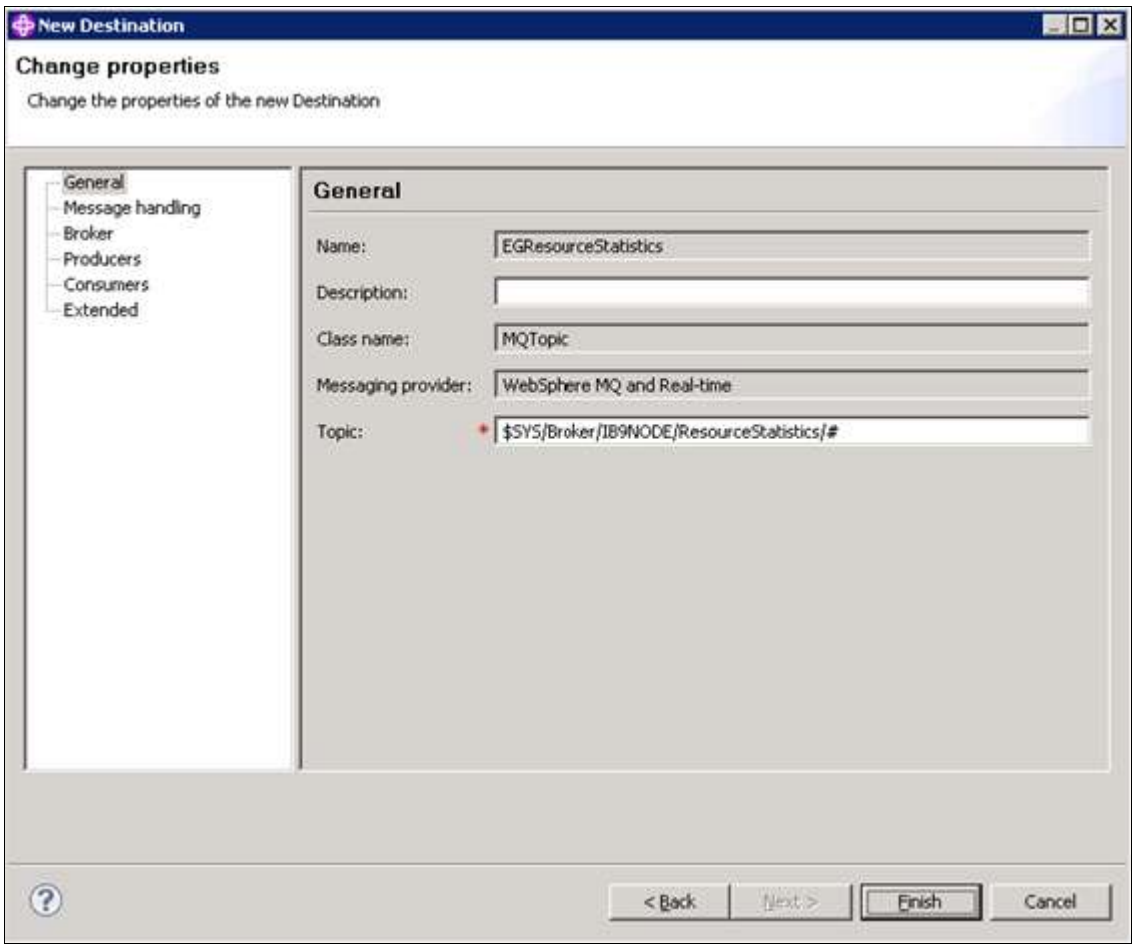

Figure 50: Providing the Topic for storing the resource statistics

4. By default, message flow statistics are stored in one destination and the resource statistics are stored in another destination. To create a destination to store the resource statistics, provide the following string in the **Topic** text box of Figure 50.

```
$SYS/Broker/<Name of the IIB node>/ResourceStatistics/#
```

In our example, the string will be as follows:

```
$SYS/Broker/IB9NODE/ResourceStatistics/#
```

5. Clicking the **Finish** button in Figure 50 will create the desired JMS destination.
6. In order to create a new destination named **EGStatisticsAccounting**, follow the steps 1-3 mentioned above. In the **Change properties** page as shown in Figure 51, provide the following string in the **Topic** text box.

```
$SYS/Broker/<name of the IIB node>/StatisticsAccounting/#
```

In our example, the string will be

```
$SYS/broker/IB9NODE/StatisticsAccounting/#
```

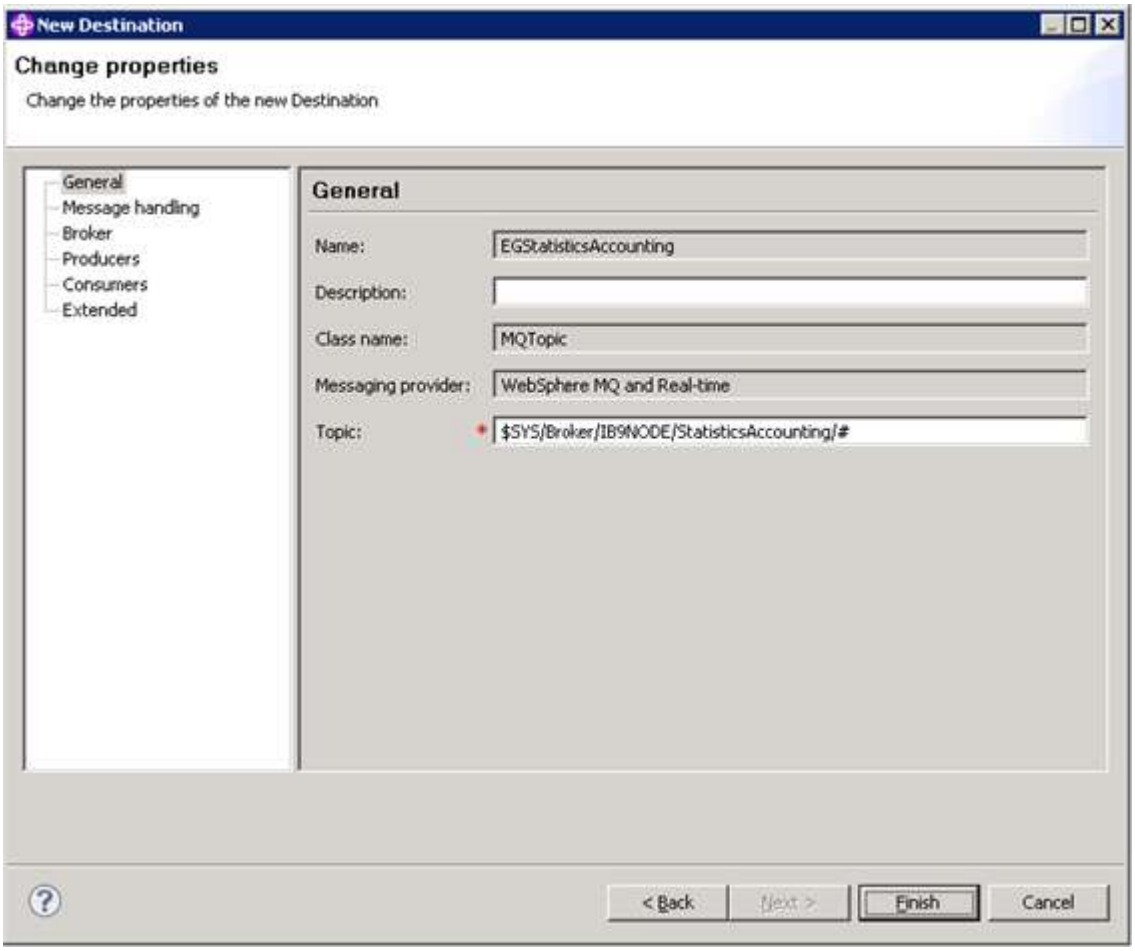

Figure 51: Providing the Topic for collecting the message flow statistics

7. Clicking the **Finish** button in Figure 51 will create the JMS destination successfully.

1.1.5 How to enable/disable the statistics accumulation in the IIB server?

To enable/disable the publication of the statistics i.e., the message flow and the resource statistics, you will be required to run the following commands from the IBM Integration Console.

1.1.5.1 Enabling the message flow statistics:

To enable the message flow statistics, run the following command from the IBM Integration Console:

mqsichangeflowstats <name of the IIB node to be monitored> -s -g -j -n advanced -t basic -b basic -c active -o xml

In our example the command will be as follows (see Figure 52):

mqsichangeflowstats IB9NODE -s -g -j -n advanced -t basic -b basic -c active -o xml

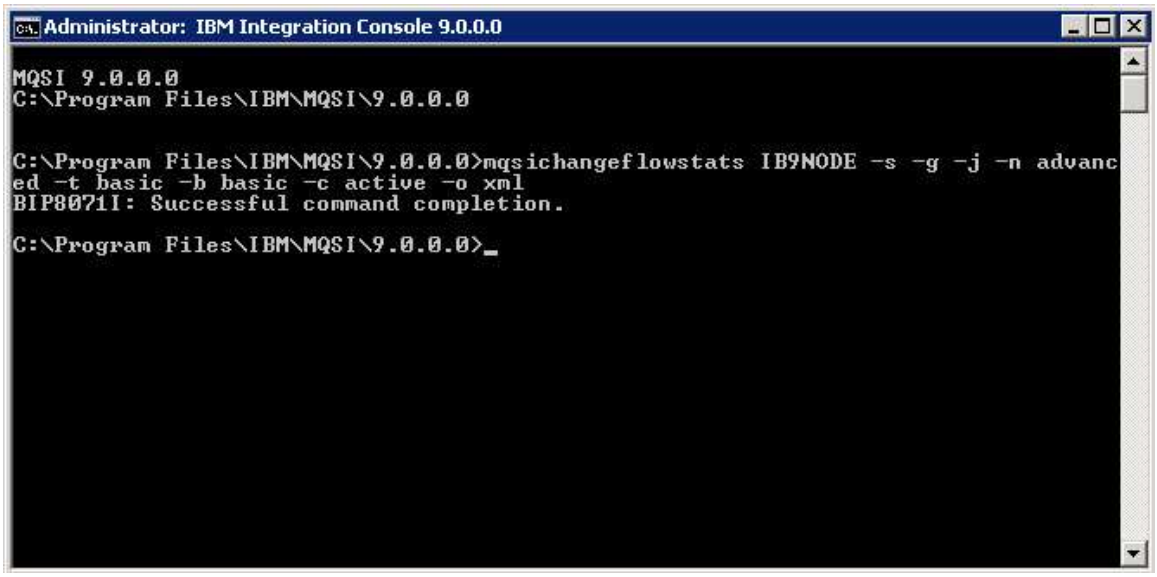

Figure 52: Enabling the message flow statistics

1.1.5.2 Disabling the message flow statistics

To disable the message flow statistics, run the following command from the IBM Integration Console:

mqsichangeflowstats <name of the IIB node to be monitored> -s -g -j -n advanced -t basic -b basic -c inactive -o xml

In our example the command will be as follows (see Figure 53):

mqsichangeflowstats IB9NODE -s -g -j -n advanced -t basic -b basic -c inactive -o xml

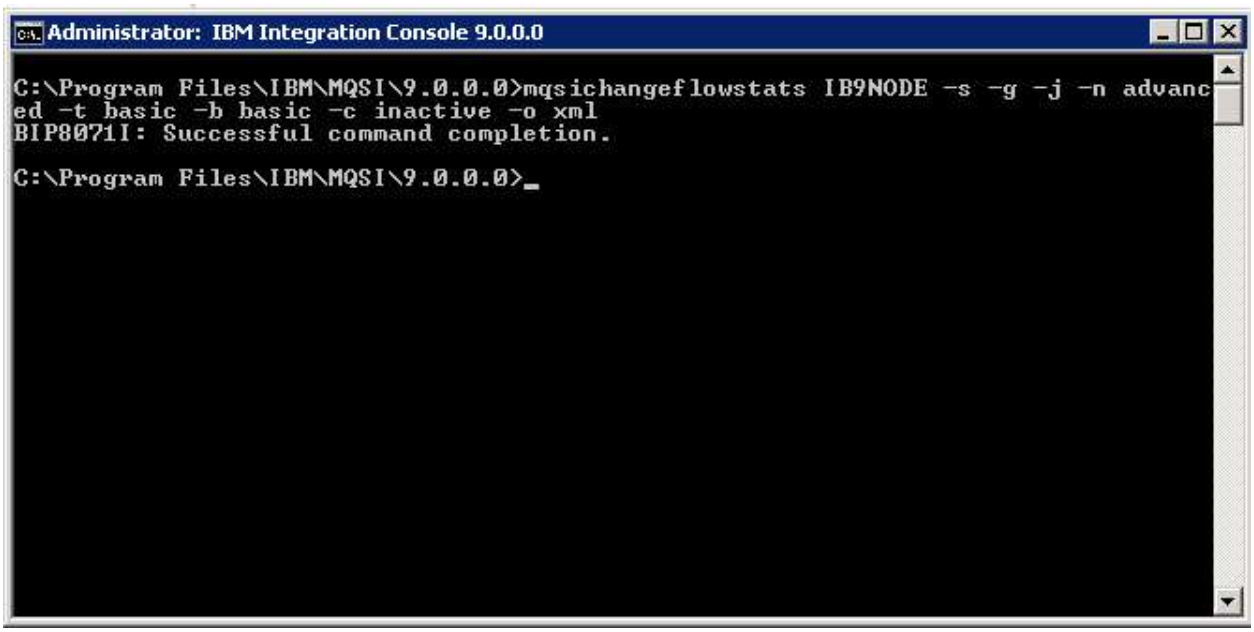

Figure 53: Disabling the message flow statistics

1.1.5.3 Enabling the resource statistics:

To enable the resource statistics, run the following command from the IBM Integration Console:

mqsichangeresourcestats <name of the IIB node to be monitored> -c active

In our example, the command to be executed is as follows (see Figure 54):

mqsichangeresourcestats IB9NODE –c active

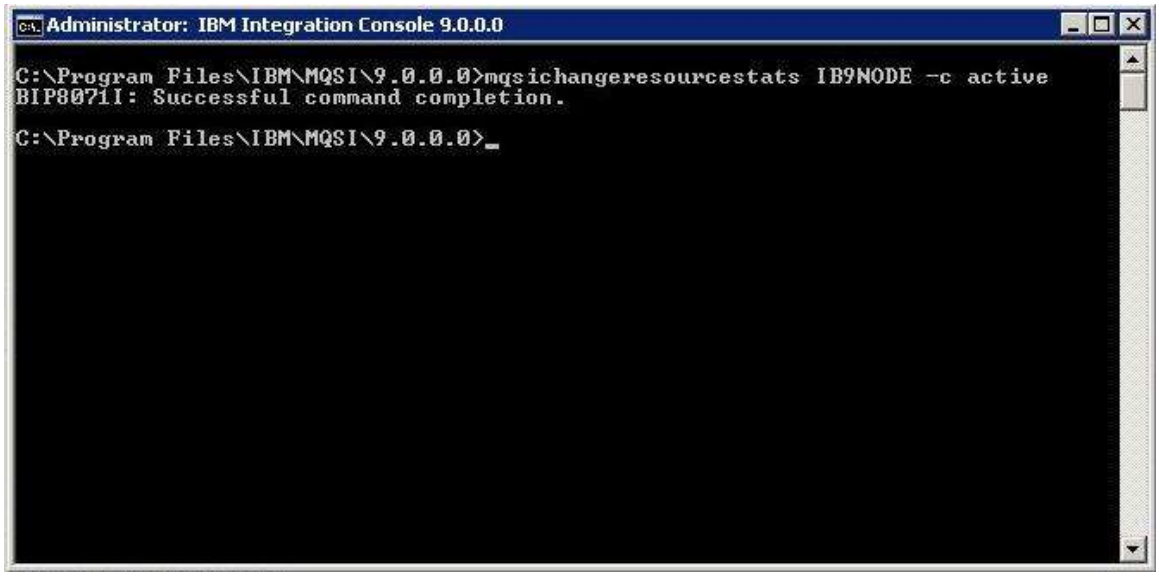

Figure 54: Enabling the resource statistics

1.1.5.4 Disabling the resource statistics

To diable the resource statistics, run the following command from the IBM Integration Console:

mqsichangeresourcestats <name of the IIB node to be monitored> -c inactive

In our example, the command to be executed is as follows (see Figure 55):

mqsichangeresourcestats IB9NODE –c inactive

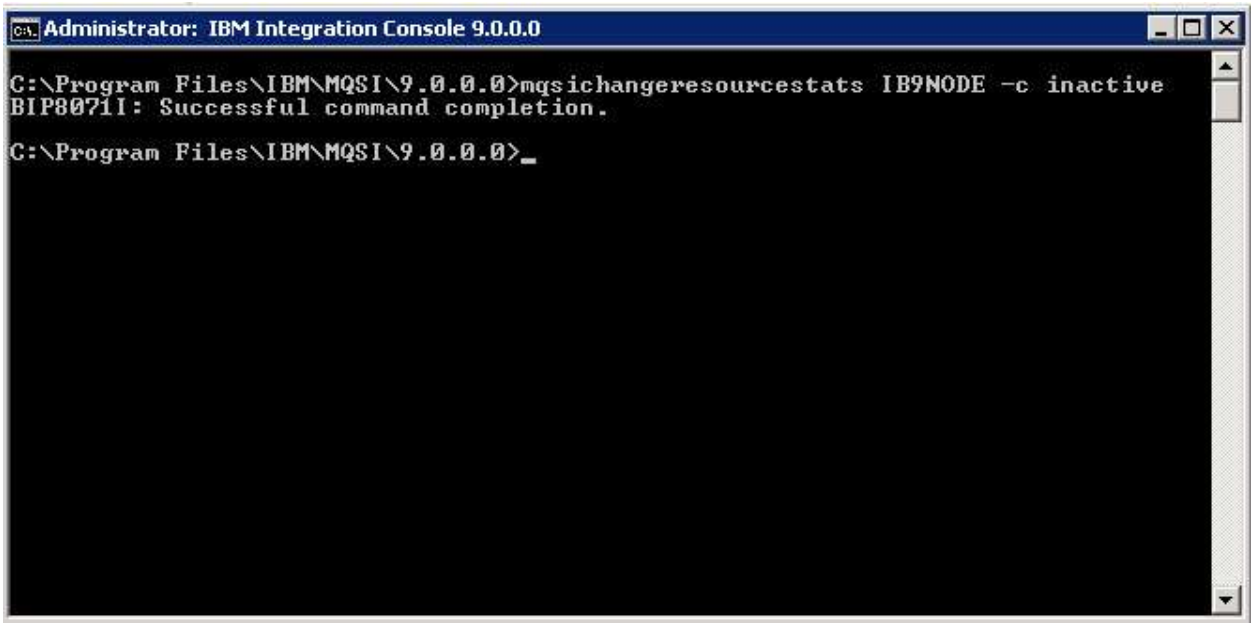

Figure 55: Disabling the resource statistics

Once you have the details of the Connection factory, JMS destinations and the Provider URL, you can proceed on to configure the tests to monitor the IBM Integration Bus. The next chapter deals exclusively on the monitoring model of the IBM Integration Bus.

2 Monitoring the IBM Integration Bus

eG Enterprise provides a specialized *IBM Integration Bus* monitoring model (see Figure 47), which periodically collects measures from various components of the IBM Integration Bus and notifies administrators of potential performance issues and processing slowdowns experienced on the whole by the IBM Integration Bus server.

Figure 56: The layer model of the IBM Integration Bus

Using the metrics reported , administrators can find quick and accurate answers for the following performance questions:

- How many requests to the FTP server were successful and how many requests failed?
- How many connections are currently available in the JBDC connection pool and how many connections are utilized?
- How many JDBC connections timed out?
- How many times the statements were executed successfully for each ODBC DSN and how many actually failed?

- How many connections were error prone for each ODBC DSN?
- What is the throughput of the messages through the SOAP service?
- How many faulty replies were sent through the SOAP service?
- Is there a resource contention on the firewall device? Which resource is bottlenecked – CPU or memory?
- How many connections can the firewall service? Is the number of connections currently handled by the firewall unusually high?
- How many connections were open on each TCP client node and TCP server node?
- How well data and messages were transmitted/received on each TCP client node and TCP server node?
- How many requests to the CICS Transaction server were successful and how many actually failed?
- How many decisions were processed successfully and how many decisions failed for each Decision service?
- How well the garbage collection activity is performed on the heap?
- How well the global cahe processes the requests and how well the global cache is utilized?
- How well each parser type processing the requests?
- How long does the message flow node take to process the input messages?
- How many messages are processed by each mssage flow node and what is the maximum time taken to process an input message?
- How long does a message flow take to process the input messages?
- How many messages are processed by each message flow and the maximum time taken to process an input message?
- How well each thread in the execution group prcesses the input messages and how long does it take to process the input messages?
- What is the processing rate of the messages for each thread?

The tests pertaining to the **IIB Application Process** layer is similar to that of the **Application Process** layer which is already dealt in the *Monitoring Unix and Windows Servers* document and the tests pertaining to the **Operating System**, **TCP** and **Network** layer have also been dealt with in the *Monitoring Unix and Windows Servers* document, Section 2.1 focuses on the **IIB Server** layer.

2.1 The IIB Server Layer

This layer tracks the specific error patterns by monitoring the execution group log files and reports the number of error patterns logged in it.

Figure 57: The tests mapped to the IIB Server layer

2.1.1 Execution group log monitor Test

This test monitors the execution group log files of the IBM Integration Bus for specific error patterns and reports administrators on the number of error patterns added to the log file.

Purpose	Monitors the execution group log files of the IBM Integration Bus for specific error patterns and reports administrators on the number of error patterns added to the log file
Target of the test	An IBM Integration Bus
Agent deploying the test	An internal agent

Configurable parameters for the test	1. **TEST PERIOD** - How often should the test be executed 2. **HOST** - The host for which the test is to be configured. 3. **PORT** – The port at which the server listens 4. **WORKPATH -** Specify the path to the log file to be monitored. For eg., *D:|zdm|logs|errorlog.* Multiple log file paths can be provided as a comma-separated list - eg., *D:|zdm|logs|errorlog,D:|zdm|logs|warnlog*. Also, instead of a specific log file path, the path to the directory containing log files can be provided - eg., *D:|zdm|logs*. This ensures that eG Enterprise monitors the most recent log files in the specified directory. Specific log file name patterns can also be specified. For example, to monitor the latest log files with names containing the strings 'error' and 'warn', the parameter specification can be, *D:|zdm|logs|*error*,D:|zdm|logs|*warn**. Here, '*' indicates leading/trailing characters (as the case may be). In this case, the eG agent first enumerates all the log files in the specified path that match the given pattern, and then picks only the latest log file from the result set for monitoring. Your **WORKPATH** specification can also be of the following format: *Name@logfilepath_or_pattern*. Here, *Name* represents the display name of the path being configured. Accordingly, the parameter specification for the 'error' and 'warn' example discussed above can be: *errors@ D:|zdm|logs|*error*,warning@D:|zdm|logs|*warn**. In this case, the display names 'error' and 'warn' will alone be displayed as descriptors of this test. **Note:** If your **WORKPATH** specification consists of file patterns that include wildcard characters (eg., *D:|zdm|logs|*error*,D:|zdm|logs|*warn**), then such configurations will only be supported in the ANSI format, and not the UTF format. Every time this test is executed, the eG agent verifies the following: Whether any changes have occurred in the size and/or timestamp of the log files that were monitoring during the last measurement period; Whether any new log files (that match the **WORKPATH** specification) have been newly added since the last measurement period; If a few lines have been added to a log file that was monitored previously, then the eG agent monitors the additions to that log file, and then proceeds to monitor newer log files (if any). If an older log file has been overwritten, then, the eG agent monitors this log file completely, and then proceeds to monitor the newer log files (if any). 5. **ROTATINGFILE** - This flag governs the display of descriptors for this test in the eG monitoring console. If this flag is set to **true** and the **WORKPATH** text box contains the full path to a specific (log/text) file, then, the descriptors of this test will be displayed in the following format: *Directory_containing_monitored_file:<SearchPattern>*. For instance, if the **WORKPATH** parameter is set to *c:|eGurkha|logs|syslog.txt*, and **ROTATINGFILE** is set to **true**, then, your descriptor will be of the following format: *c:|eGurkha|logs:<SearchPattern>*. On the other hand, if the **ROTATINGFILE** flag had been set to **false**, then the descriptors will be of the following format: *<FileName>:<SearchPattern>* - i.e., *syslog.txt:<SearchPattern>* in the case of the example above.

If this flag is set to **true** and the **WORKPATH** parameter is set to the directory containing log files, then, the descriptors of this test will be displayed in the format: *Configured_directory_path:<SearchPattern>.* For instance, if the **WORKPATH** parameter is set to *c:|eGurkha|logs*, and **ROTATINGFILE** is set to **true**, then, your descriptor will be: *c:|eGurkha|logs:<SearchPattern>*. On the other hand, if the **ROTATINGFILE** parameter had been set to **false**, then the descriptors will be of the following format: *Configured_directory:<SearchPattern>* - i.e., *logs:<SearchPattern>* in the case of the example above.

If this flag is set to true and the **WORKPATH** parameter is set to a specific file pattern, then, the descriptors of this test will be of the following format: *<FilePattern>:<SearchPattern>*. For instance, if the **WORKPATH** parameter is set to *c:|eGurkha|logs|*sys**, and **ROTATINGFILE** is set to **true**, then, your descriptor will be: **sys*:<SearchPattern>*. In this case, the descriptor format will not change even if the **ROTATINGFILE** flag status is changed .**DD FREQUENCY** - Refers to the frequency with which detailed diagnosis measures are to be generated for this test. The default is *1:1*. This indicates that, by default, detailed measures will be generated every time this test runs, and also every time the test detects a problem. You can modify this frequency, if you so desire. Also, if you intend to disable the detailed diagnosis capability for this test, you can do so by specifying *none* against **DD FREQUENCY**.

6. **SEARCHPATTERN** - Enter the specific patterns of alerts to be monitored. The pattern should be in the following format: *<PatternName>:<Pattern>*, where *<PatternName>* is the pattern name that will be displayed in the monitor interface and *<Pattern>* is an expression of the form - *expr* or expr or *expr or expr*, etc. A leading '*' signifies any number of leading characters, while a trailing '*' signifies any number of trailing characters.

   For example, say you specify *error:error-** in the **SEARCHPATTERN** text box. This indicates that "error" is the pattern name to be displayed in the monitor interface. "error-*" indicates that the test will monitor only those lines in the alert log which start with the term "error-".

   A single pattern may also be of the form e1+e2, where + signifies an OR condition. That is, the <*PatternName*> is matched if either e1 is true or e2 is true.

   Multiple search patterns can be specified as a comma-separated list. For example: *error:error-*,offline:*offline*,online:*online*

   If the **WORKPATH** specification is of the format *Name@logfilepath*, then the descriptor for this test in the eG monitor interface will be of the format: *Name:PatternName*. On the other hand, if the **WORKPATH** specification consists only of a comma-separated list of log file paths, then the descriptors will be of the format: *LogFilePath:PatternName*.

   If you want all the messages in a log file to be monitored, then your specification would be: *<PatternName>:**.

7. **LINES** - Specify two numbers in the format x:y. This means that when a line in the alert file matches a particular pattern, then x lines before the matched line and y lines after the matched line will be reported in the detailed diagnosis output (in addition to the matched line). The default value here is 0:0. Multiple entries can be provided as a comma-separated list.

   If you give 1:1 as the value for **LINES**, then this value will be applied to all the patterns specified in the **SEARCHPATTERN** field. If you give 0:0,1:1,2:1 as the value for **LINES** and if the corresponding value in the **SEARCHPATTERN** filed is like error:error-*,offline:*offline*,online:*online then:

0:0 will be applied to error:error-* pattern

1:1 will be applied to offline:*offline* pattern

2:1 will be applied to online:*online pattern

8. **EXCLUDEPATTERN** - Provide a comma-separated list of patterns to be excluded from monitoring in the **EXCLUDEPATTERN** text box. For example **critical*, *exception**. By default, this parameter is set to 'none'.

9. **UNIQUEMATCH** - By default, the **UNIQUEMATCH** parameter is set to **FALSE**, indicating that, by default, the test checks every line in the log file for the existence of each of the configured **SEARCHPATTERNS**. By setting this parameter to **TRUE**, you can instruct the test to ignore a line and move to the next as soon as a match for one of the configured patterns is found in that line. For example, assume that *Pattern1:*fatal*,Pattern2:*error** is the **SEARCHPATTERN** that has been configured. If **UNIQUEMATCH** is set to **FALSE**, then the test will read every line in the log file completely to check for the existence of messages embedding the strings 'fatal' and 'error'. If both the patterns are detected in the same line, then the number of matches will be incremented by 2. On the other hand, if **UNIQUEMATCH** is set to **TRUE**, then the test will read a line only until a match for one of the configured patterns is found and not both. This means that even if the strings 'fatal' and 'error' follow one another in the same line, the test will consider only the first match and not the next. The match count in this case will therefore be incremented by only 1.

10. **CASESENSITIVE** - This flag is set to **No** by default. This indicates that the test functions in a 'case-insensitive' manner by default. This implies that, by default, the test ignores the case of your **WORKPATH** and **SEARCHPATTERN** specifications. If this flag is set to **Yes** on the other hand, then the test will function in a 'case-sensitive' manner. In this case therefore, for the test to work, even the case of your **WORKPATH** and **SEARCHPATTERN** specifications should match with the actuals.

11. **ROLLOVERFILE** - By default, this flag is set to **false**. Set this flag to **true** if you want the test to support the 'roll over' capability of the specified **WORKPATH**. A roll over typically occurs when the timestamp of a file changes or when the log file size crosses a pre-determined threshold. When a log file rolls over, the errors/warnings that pre-exist in that file will be automatically copied to a new file, and all errors/warnings that are captured subsequently will be logged in the original/old file. For instance, say, errors and warnings were originally logged to a file named *error_log*. When a roll over occurs, the content of the file *error_log* will be copied to a file named *error_log.1*, and all new errors/warnings will be logged in *error_log*. In such a scenario, since the **ROLLOVERFILE** flag is set to **false** by default, the test by default scans only *error_log.1* for new log entries and ignores *error_log*. On the other hand, if the flag is set to **true**, then the test will scan both *error_log* and *error_log.1* for new entries.

    If you want this test to support the 'roll over' capability described above, the following conditions need to be fulfilled:

    - The **WORKPATH** parameter has to be configured only with the name and/or path of one/more alert files. File patterns or directory specifications should not be specified in the **WORKPATH** text box.

    - The roll over file name should be of the format: "<**WORKPATH**>.1", and this file must be in the same directory as the **WORKPATH**.

12. **OVERWRITTENFILE** - By default, this flag is set to **false**. Set this flag to **true** if log files do not 'roll over' in your environment, but get overwritten instead. In such environments typically, new error/warning messages that are captured will be written into the log file that pre-exists and will replace the original contents of that log file; unlike when 'roll over' is enabled, no new log files are created for new entries in this case. If the **OVERWRITTENFILE** flag is set to **true**, then the test will scan the new entries in the log file for matching patterns. However, if the flag is set to **false**, then the test will ignore the new entries.

13. **ENCODEFORMAT** – By default, this is set to *none*, indicating that no encoding format applies by default. However, if the test has to use a specific encoding format for reading from the specified **WORKPATH** , then you will have to provide a valid encoding format here - eg., *UTF-8, UTF-16*, etc. Where multiple log files are being monitored, you will have to provide a comma-separated list of encoding formats – one each for every log file monitored. Make sure that your encoding format specification follows the same sequence as your **WORKPATH** specification. In other words, the first encoding format should apply to the first alert file, and so on. For instance, say that your alertfile specification is as follows: *D:|logs|report.log,E:|logs|error.log, C:|logs|warn_log*. Assume that while *UTF-8* needs to be used for reading from *report.log* , *UTF-16* is to be used for reading from *warn_log* . No encoding format need be applied to *error*.log. In this case, your **ENCODEFORMAT** specification will be: *UTF-8,none,UTF-16*.

14. **USEUTF8** - If UTF-8 encoding is to be used for reading the specified log file, then, set the **USEUTF8** flag to *true*. By default, this flag is set to *false*. If multiple log files are being monitored, then, for each file, you will have to indicate whether UTF-8 encoding is to be used for reading that file or not. For instance, assume that the **WORKPATH** parameter is set to *errors@d:|zdm|logs|error.log,warnings@d:|zdm|logs|warn.log* Now, to instruct the test to use UTF-8 encoding for reading the 'errors' log file and not to use the UTF-8 encoding while reading the 'warnings' log file, your **USEUTF8** setting should be as follows: *true,false*. **Note that the number of values provided against the USEUTF8 parameter should be equal to the number of log files being monitored. Also, note that if the ALERTFILE being monitored has BOM, then the test will automatically use UTF-8 encoding to read that file, even if the USEUTF8 flag is set to false.**

> **Note:**
>
> If your **WORKPATH** specification consists of file patterns that include wildcard characters (eg *d:|zdm|logs|*error*,d:|zdm|logs|*warn**), then the files that match such patterns will only support the ANSI format, and not the UTF format, even if the **UTF-8** parameter is set to **true** for such patterns.

15. **USEUTF16 -** If UTF-16 encoding is to be used for reading the specified log file, then, set the **USEUTF16** flag to **true**. By default, this flag is set to **true**. If multiple log files are being monitored, then, for each file, you will have to indicate whether UTF-16 encoding is to be used for reading that file or not. For instance, assume that the **WORKPATH** parameter is set to *soaplog@"C:|ProgramData|Application Data|IBM|*|Log|*soap*",conlogs@"C:|ProgramData|Application Data|IBM|Log|*con*"*. Now, to instruct the test to use UTF-16 encoding for reading the 'soaplog' log file and not to use the UTF-16 encoding while reading the 'conlogs' log file, your **USEUTF16** setting should be as follows: *true,false*. **Note that the number of values provided against the USEUTF8 parameter should be equal to the number of log files being monitored.**

<table>
<tr><td></td><td colspan="3">16. DD FREQUENCY - Refers to the frequency with which detailed diagnosis measures are to be generated for this test. The default is 1:1. This indicates that, by default, detailed measures will be generated every time this test runs, and also every time the test detects a problem. You can modify this frequency, if you so desire. Also, if you intend to disable the detailed diagnosis capability for this test, you can do so by specifying none against DD FREQUENCY.
17. DETAILED DIAGNOSIS - To make diagnosis more efficient and accurate, the eG Enterprise suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the On option. To disable the capability, click on the Off option.
The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled:
• The eG manager license should allow the detailed diagnosis capability
• Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.</td></tr>
<tr><td>ll Outputs of the test</td><td colspan="3">One set of results for every WORKPATH and SEARCHPATTERN combination</td></tr>
<tr><td>Measurements made by the test</td><td>Measurement</td><td>Measurement Unit</td><td>Interpretation</td></tr>
<tr><td></td><td>Number of messages:
Indicates the number of messages of the configured patterns that were added to the execution group logs when the test was last executed.</td><td>Number</td><td>The value of this measure is a clear indicator of the number of "new" alerts that have come into the monitored logs.

The detailed diagnosis of this measure of enabled, lists the recent messages.</td></tr>
</table>

2.2 The IIB Protocol Resources Layer

This layer helps administrators to identify the load processing capability of the FTP server integrated with the IIB server, the connections utilized by the JDBC Provider configurable service, the numerical statistics of the connections for each ODBC DSN, the load on the SOAP service, the TCP connection s to and from the TCP client node and the TCP server node. Using the tests of this layer, administrators can be proactively alerted to bottlenecks/slowdowns in the processing capability of the IIB server.

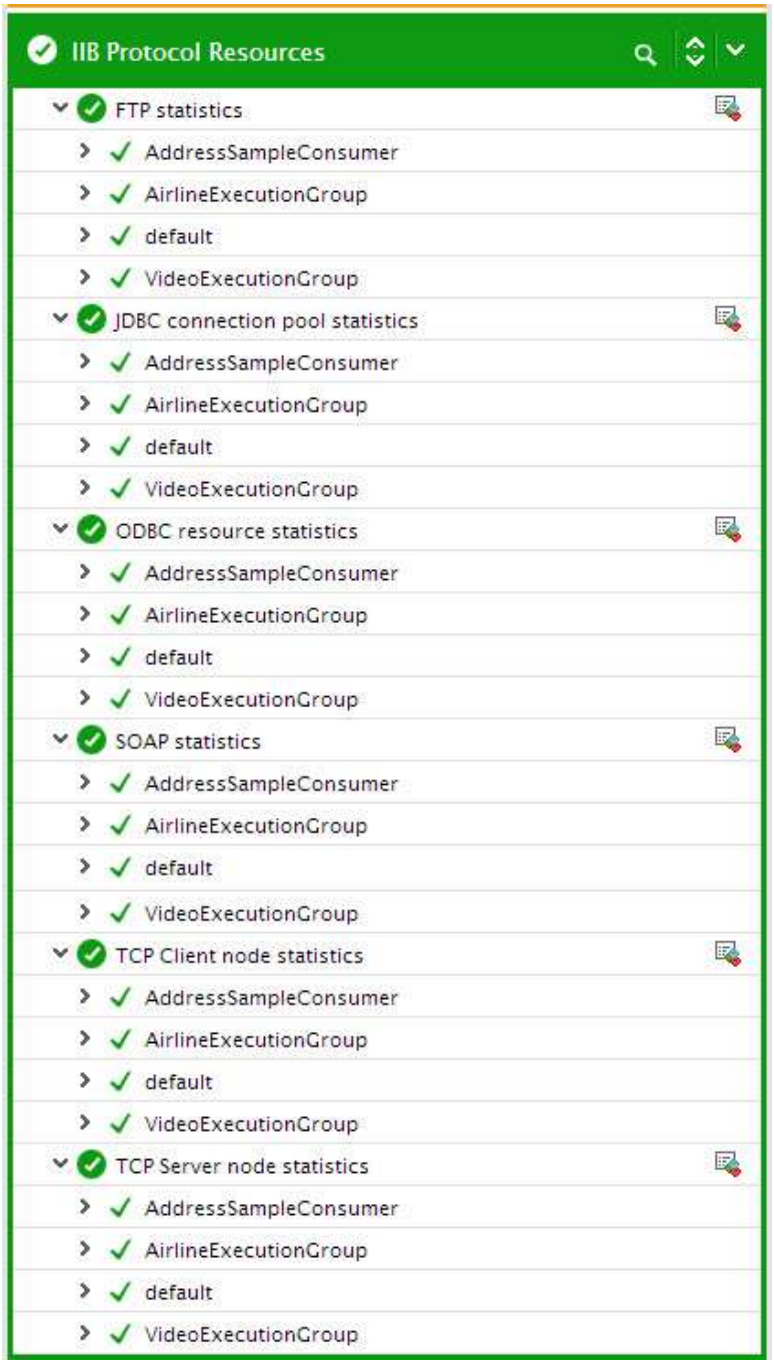

Figure 58: The tests mapped to the IIB Protocol Resources layer

2.2.1 FTP Statistics Test

The File Transfer Protocol (FTP) is a standard network protocol used to transfer computer files from one host to another host over a TCP-based network, such as the Internet.

This test auto discovers the FTP servers integrated with the IBM Integration Bus and reports the current load on the FTP server. This way, administrators may be proactively alerted to load processing bottlenecks.

Purpose	Auto discovers the FTP servers integrated with the IBM Integration Bus and reports the current load on the FTP server. This way, administrators may be proactively alerted to load processing bottlenecks.		
Target of the test	An IBM Integration Bus		
Agent deploying the test	An internal/remote agent		
Configurable parameters for the test	1. **TEST PERIOD** - How often should the test be executed 2. **HOST** - The host for which the test is to be configured 3. **PORT** – The port on which the specified host listens 4. **JNDI NAMESPACE LOCATION** – Indicate where the JNDI namespace has been created. For this, select one of the following options: **File** or **LDAP**. Select **File** if the JNDI namespace is created in the same location as that of the IIB server. 5. **SSL** - If you have chosen **LDAP** as the **JNDI NAMESPACE LOCATION** wherein the LDAP server communicating with the IIB server that is to be monitored is an SSL-enabled server, then set the **SSL** flag to **Yes**. By default, this flag is set to **No**. 6. **USER DN** and **PASSWORD** – Here, provide the credentials of the Active Directory user created as explained in Section 1.1.4.1, if the target IIB server is monitored in an agentless manner. If the IIB server is to be monitored in an agent based manner, specify *none* against these parameters. 7. **CONFIRM PASSWORD** – Confirm the **PASSWORD** by retyping it in this text box. 8. **JNDI PROVIDER URL** - Specify the URL that was specified as the provider URL while creating the initial context. To know the Provider URL, refer to Figure 7 if you choose to monitor the target IIB server in an agent based manner and Figure 34 if you chose to monitor the target IIB server in an agentless manner. 9. **TOPICCONNECTIONFACTORY** – Specify the name of the connection factory in this text box. If you have chosen to monitor the target IIB server in an agentless manner, refer to Section 1.1.3.2.1 and if you have chosen to monitor the target IIB server in an agentless manner, refer to Section 1.1.4.3.1. 10. **JMS RESOURCE STATS** – Specify the name of the JMS destination that you have created for storing the Resource statistics. If you chose to monitor the target IIB server in an agentless manner, refer to Section 1.1.4.3.2 and if you choose to monitor the target IIB server in an agent based manner, refer to Section 1.1.3.2.2. 11. **JMS FLOW STATS** – Specify the name of the JMS destination that you have created for storing the Message flow statistics. If you chose to monitor the target IIB server in an agentless manner, refer to Section 1.1.4.3.2 and if you choose to monitor the target IIB server in an agent based manner, refer to Section 1.1.3.2.2.		
ll Outputs of the test	One set of results for each *Execution group:FTP server* integrated with the IBM Integration Bus that is to be monitored		
Measurements made by the	**Measurement**	**Measurement Unit**	**Interpretation**

test	**FTP gets rate:** Indicates the rate at which transfers were made from this FTP server to the file system of the Integration node during the last measurement period.	Number/sec	
	Data received rate: Indicates the rate at which data is received by the file system of the integration node from this FTP server during the last measurement period.	Bytes/sec	Compare the value of this measure across FTP servers to identify the server that is the most busy in transferring the data. This measure is a good indicator of the load on the FTP server.
	FTP puts rate: Indicates the rate at which transfers were made to this remote server from the file system of the Integration node during the last measurement period.	Number/sec	
	Data sent rate: Indicates the rate at which data is transferred to this FTP server from the file system of the Integration node during the last measurement period.	Bytes/sec	Compare the value of this measure across FTP servers to identify the server that is busy receiving data.

2.2.2 JDBC connection pool statistics Test

The Java Database Connectivity (JDBC) is an industry standard for database-independent connectivity between the Java platform and a wide range of databases. The JDBC interface provides a call-level API for SQL-based and XQuery-based database access.

This test auto discovers the JDBC Provider configurable service of the IBM Integration Bus and reports how well connections are utilized by each service in the connection pool. Using this test, you can figure out the number of connections that are available for use and the requests that are handled by the connection pool. This test proactively alerts administrators to slowdowns/bottlenecks in obtaining connections and the timeouts experienced while waiting for connections.

Purpose	Auto discovers the JDBC Provider configurable service of the IBM Integration Bus and reports how well connections are utilized by each service in the connection pool. Using this test, you can figure out the number of connections that are available for use and the requests that are handled by the connection pool. This test proactively alerts administrators to slowdowns/bottlenecks in obtaining connections and the timeouts experienced while waiting for connections
Target of the test	An IBM Integration Bus

Agent deploying the test	An internal/remote agent		
Configurable parameters for the test	1. **TEST PERIOD** - How often should the test be executed 2. **HOST** - The host for which the test is to be configured 3. **PORT** – The port on which the specified host listens 4. **JNDI NAMESPACE LOCATION** – Indicate where the JNDI namespace has been created. For this, select one of the following options: **File** or **LDAP**. Select **File** if the JNDI namespace is created in the same location as that of the IIB server. 5. **SSL** - If you have chosen **LDAP** as the **JNDI NAMESPACE LOCATION** wherein the LDAP server communicating with the IIB server that is to be monitored is an SSL-enabled server, then set the **SSL** flag to **Yes**. By default, this flag is set to **No**. 6. **USER DN** and **PASSWORD** – Here, provide the credentials of the Active Directory user created as explained in Section 1.1.4.1, if the target IIB server is monitored in an agentless manner. If the IIB server is to be monitored in an agent based manner, specify *none* against these parameters. 7. **CONFIRM PASSWORD** – Confirm the **PASSWORD** by retyping it in this text box. 8. **JNDI PROVIDER URL** - Specify the URL that was specified as the provider URL while creating the initial context. To know the Provider URL, refer to Figure 7 if you choose to monitor the target IIB server in an agent based manner and Figure 34 if you chose to monitor the target IIB server in an agentless manner. 9. **TOPICCONNECTIONFACTORY** – Specify the name of the connection factory in this text box. If you have chosen to monitor the target IIB server in an agentless manner, refer to Section 1.1.3.2.1 and if you have chosen to monitor the target IIB server in an agentless manner, refer to Section 1.1.4.3.1. 10. **JMS RESOURCE STATS** – Specify the name of the JMS destination that you have created for storing the Resource statistics. If you chose to monitor the target IIB server in an agentless manner, refer to Section 1.1.4.3.2 and if you choose to monitor the target IIB server in an agent based manner, refer to Section 1.1.3.2.2. 11. **JMS FLOW STATS** – Specify the name of the JMS destination that you have created for storing the Message flow statistics. If you chose to monitor the target IIB server in an agentless manner, refer to Section 1.1.4.3.2 and if you choose to monitor the target IIB server in an agent based manner, refer to Section 1.1.3.2.2.		
Outputs of the test	One set of results for each *Execution group:JDBC Provider configurable service* that is using connection pooling in the IBM Integration Bus that is to be monitored		
Measurements made by the	**Measurement**	**Measurement Unit**	**Interpretation**

test	**Current connection in the pool:** Indicates the number of connections that are currently available in the connection pool utilized by this service.	Number	
	Connection pool usage: Indicates the percentage of connections that were utilized in the connection pool during the last measurement period.	Percent	A value close to 100 indicates an overload condition.
	Free connections in the pool: Indicates the number of connections that were available for use in the connection pool during the last measurement period.	Number	A high value is desired for this measure. A gradual/sudden increase in the value of this measure indicates that the connection pool is currently overloaded.
	Requests handled by the pool: Indicates the rate at which requests were handled by the connection pool during the last measurement period.	Requests/sec	
	Connection wait requests: Indicates the rate at which requests for a connection could not be satisfied immediately.	Requests/sec	A high value for this measure indicates that the connection pool is overloaded. A request could not be satisified when the number of allocated connections reached the maximum connection pool size and there are no connections currently available.
	Connection timed out requests: Indicates the rate at which the requests for connections could not be satisfied by this connection pool within 15 seconds during the last measurement period.	Requests/sec	Ideally, the value of this measure should be zero.
	Maximum delay: Indicates the maximum time taken to allocate a connection in the connection pool during the last measurement period.	Millisecs	A high value for this measure is a cause of concern.

2.2.3 ODBC resource statistics Test

The Open Database Connectivity (ODBC) is a standard application programming interface (API) used for accessing data in both relational and non-relational database management systems. By using this API, database applications can access data stored in database management systems on various computers, even if each database management system uses a different data storage format and programming interface. Often administrators may want to figure put how well the ODBC DSN is utilized in the target environment. The **ODBC resource statistics** test exactly helps administrators identify the same! This test auto discovers the ODBC DSNs integrated with the IBM Integration Bus and for each ODBC DSN, reports the numerical statistics of the following:

- Successful execution of the statements
- Statements that failed execution
- Active connections
- Closed connections
- Connections that are experiencing errors etc

Purpose	Auto discovers the ODBC DSNs integrated with the IMB Integration Bus and for each ODBC DSN, reports the numerical statistics of the following: ➢ Successful execution of the statements ➢ Statements that failed execution ➢ Active connections ➢ Closed connections ➢ Connections that are experiencing errors etc
Target of the test	An IBM Integration Bus
Agent deploying the test	An internal/remote agent

<table>
<tr><td>Configurable parameters for the test</td><td colspan="3">1. TEST PERIOD - How often should the test be executed
2. HOST - The host for which the test is to be configured
3. PORT – The port on which the specified host listens
4. JNDI NAMESPACE LOCATION – Indicate where the JNDI namespace has been created. For this, select one of the following options: File or LDAP. Select File if the JNDI namespace is created in the same location as that of the IIB server.
5. SSL - If you have chosen LDAP as the JNDI NAMESPACE LOCATION wherein the LDAP server communicating with the IIB server that is to be monitored is an SSL-enabled server, then set the SSL flag to Yes. By default, this flag is set to No.
6. USER DN and PASSWORD – Here, provide the credentials of the Active Directory user created as explained in Section 1.1.4.1, if the target IIB server is monitored in an agentless manner. If the IIB server is to be monitored in an agent based manner, specify none against these parameters.
7. CONFIRM PASSWORD – Confirm the PASSWORD by retyping it in this text box.
8. JNDI PROVIDER URL - Specify the URL that was specified as the provider URL while creating the initial context. To know the Provider URL, refer to Figure 7 if you choose to monitor the target IIB server in an agent based manner and Figure 34 if you chose to monitor the target IIB server in an agentless manner.
9. TOPICCONNECTIONFACTORY – Specify the name of the connection factory in this text box. If you have chosen to monitor the target IIB server in an agentless manner, refer to Section 1.1.3.2.1 and if you have chosen to monitor the target IIB server in an agentless manner, refer to Section 1.1.4.3.1.
10. JMS RESOURCE STATS – Specify the name of the JMS destination that you have created for storing the Resource statistics. If you chose to monitor the target IIB server in an agentless manner, refer to Section 1.1.4.3.2 and if you choose to monitor the target IIB server in an agent based manner, refer to Section 1.1.3.2.2.
11. JMS FLOW STATS – Specify the name of the JMS destination that you have created for storing the Message flow statistics. If you chose to monitor the target IIB server in an agentless manner, refer to Section 1.1.4.3.2 and if you choose to monitor the target IIB server in an agent based manner, refer to Section 1.1.3.2.2.</td></tr>
<tr><td>Outputs of the test</td><td colspan="3">One set of results for each ODBC DSN integrated with the IBM Integration Bus that is to be monitored</td></tr>
<tr><td rowspan="2">Measurements made by the test</td><td>Measurement</td><td>Measurement Unit</td><td>Interpretation</td></tr>
<tr><td>Successful execution rate:
Indicates the number of times statements were executed successfully through this ODBC DSN during the last measurement period.</td><td>Number</td><td>A high value is desired for this measure.</td></tr>
</table>

	Failed execution rate: Indicates the number of times statements failed to execute through this ODBC DSN during the last measurement period.	Number	Ideally, the value of this measure should be zero.
	Total Execution rate: Indicates the total number of times statements were executed through this ODBC DSN during the last measurement period.	Number	The value of this measure is cumulative of the *Successful execution rate* and the *Failed execution rate* measures.
	Active connections: Indicates the number of connections that were open to this ODBC DSN during the last measurement period.	Number	
	Closed connections: Indicates the number of connections to this ODBC DSN that were closed during the last measurement period.	Number	The value of this measure includes the connections that were closed due to errors, connections forced to close by the DBMS and the connections closed by the broker because the connections were no longer required.
	Connection errors: Indicates the number of times the connections to this ODBC DSN were error prone during the last measurement period.	Number	A low value is desired for this measure.

2.2.4 SOAP statistics Test

SOAP is a lightweight, XML-based protocol used for exchanging information in a decentralized, distributed environment. SOAP can be used to query and return information and invoke services across the internet.

For each SOAP URL/service integrated with the target IBM Integration Bus, this test reports how well messages are sent/received from the SOAP client. This test is therefore useful for administrators to identify the load on the SOAP client and figure out the errors that occurred when messages are sent to the message flow. This way administrators may be proactively alerted to processing bottlenecks in the SOAP client.

Purpose	For each SOAP URL/service integrated with the target IBM Integration Bus, this test reports how well messages are sent/received from the SOAP client. This test is therefore useful for administrators to identify the load on the SOAP client and figure out the errors that occurred when messages are sent to the message flow.
Target of the test	An IBM Integration Bus
Agent deploying the test	An internal/remote agent

<table>
<tr><td>Configurable parameters for the test</td><td colspan="3">1. TEST PERIOD - How often should the test be executed
2. HOST - The host for which the test is to be configured
3. PORT – The port on which the specified host listens
4. JNDI NAMESPACE LOCATION – Indicate where the JNDI namespace has been created. For this, select one of the following options: File or LDAP. Select File if the JNDI namespace is created in the same location as that of the IIB server.
5. SSL - If you have chosen LDAP as the JNDI NAMESPACE LOCATION wherein the LDAP server communicating with the IIB server that is to be monitored is an SSL-enabled server, then set the SSL flag to Yes. By default, this flag is set to No.
6. USER DN and PASSWORD – Here, provide the credentials of the Active Directory user created as explained in Section 1.1.4.1, if the target IIB server is monitored in an agentless manner. If the IIB server is to be monitored in an agent based manner, specify none against these parameters.
7. CONFIRM PASSWORD – Confirm the PASSWORD by retyping it in this text box.
8. JNDI PROVIDER URL - Specify the URL that was specified as the provider URL while creating the initial context. To know the Provider URL, refer to Figure 7 if you choose to monitor the target IIB server in an agent based manner and Figure 34 if you chose to monitor the target IIB server in an agentless manner.
9. TOPICCONNECTIONFACTORY – Specify the name of the connection factory in this text box. If you have chosen to monitor the target IIB server in an agentless manner, refer to Section 1.1.3.2.1 and if you have chosen to monitor the target IIB server in an agentless manner, refer to Section 1.1.4.3.1.
10. JMS RESOURCE STATS – Specify the name of the JMS destination that you have created for storing the Resource statistics. If you chose to monitor the target IIB server in an agentless manner, refer to Section 1.1.4.3.2 and if you choose to monitor the target IIB server in an agent based manner, refer to Section 1.1.3.2.2.
11. JMS FLOW STATS – Specify the name of the JMS destination that you have created for storing the Message flow statistics. If you chose to monitor the target IIB server in an agentless manner, refer to Section 1.1.4.3.2 and if you choose to monitor the target IIB server in an agent based manner, refer to Section 1.1.3.2.2.</td></tr>
<tr><td>Outputs of the test</td><td colspan="3">One set of results for each Execution group: SOAP URL/service of the IBM Integration Bus that is to be monitored</td></tr>
<tr><td rowspan="2">Measurements made by the test</td><td>Measurement</td><td>Measurement Unit</td><td>Interpretation</td></tr>
<tr><td>Inbound message rate:
Indicates the rate at which messages are received from the SOAP client through this service during the last measurement period.</td><td>Messages/sec</td><td>Ideally, the value of this measure should be high. A consistent decrease in this value indicates that there is a delay while reading messages from the SOAP client. Further investigation may be required to diagnose the root-cause of the slowdown.</td></tr>
</table>

	Replies sent rate: Indicates the rate at which replies are sent to the SOAP client from the SOAP Reply node through this service during the last measurement period.	Replies/sec	
	Flow throughput: Indicates the rate at which messages were sent to the message flow through this service without any faults during the last measurement period.	Messages/sec	This measure is a good indicator of the load on the SOAP client.
	Inbound messages faulted before flow: Indicates the number of messages through this service that faulted before reaching the message flow during the last measurement period.	Number	Ideally, the value of this measure should be zero.
	Inbound messages faulted rate: Indicates the rate at which messages through this service faulted before reaching the message flow during the last measurement period.	Messages/sec	Ideally, the value of this emasure should be zero.
	Fault replies sent: Indicates the number of faulty replies sent through this service during the last measurement period.	Number	The faulty replies may be user defiend or broker exceptions.
	Fault replies rate: Indicates the rate at which faulty replies were sent through this service during the last measurement period.	Replies/sec	The faulty replies may be user defiend or broker exceptions.

2.2.5 TCP Client node statistics Test

This test monitors the TCP connections to the TCP client node and reports the count of the connections that were open, closed and failed. In the process, this test proactively alerts administrators to processing bottlenecks on the TCP client node.

Purpose	Monitors the TCP connections to the TCP client node and reports the count of the connections that were open, closed and failed. In the process, this test proactively alerts administrators to

	processing bottlenecks on the TCP client node		
Target of the test	An IBM Integration Bus		
Agent deploying the test	An internal/remote agent		
Configurable parameters for the test	1. **TEST PERIOD** - How often should the test be executed 2. **HOST** - The host for which the test is to be configured 3. **PORT** – The port on which the specified host listens 4. **JNDI NAMESPACE LOCATION** – Indicate where the JNDI namespace has been created. For this, select one of the following options: **File** or **LDAP**. Select **File** if the JNDI namespace is created in the same location as that of the IIB server. 5. **SSL** - If you have chosen **LDAP** as the **JNDI NAMESPACE LOCATION** wherein the LDAP server communicating with the IIB server that is to be monitored is an SSL-enabled server, then set the **SSL** flag to **Yes**. By default, this flag is set to **No**. 6. **USER DN** and **PASSWORD** – Here, provide the credentials of the Active Directory user created as explained in Section 1.1.4.1, if the target IIB server is monitored in an agentless manner. If the IIB server is to be monitored in an agent based manner, specify *none* against these parameters. 7. **CONFIRM PASSWORD** – Confirm the **PASSWORD** by retyping it in this text box. 8. **JNDI PROVIDER URL** - Specify the URL that was specified as the provider URL while creating the initial context. To know the Provider URL, refer to Figure 7 if you choose to monitor the target IIB server in an agent based manner and Figure 34 if you chose to monitor the target IIB server in an agentless manner. 9. **TOPICCONNECTIONFACTORY** – Specify the name of the connection factory in this text box. If you have chosen to monitor the target IIB server in an agentless manner, refer to Section 1.1.3.2.1 and if you have chosen to monitor the target IIB server in an agentless manner, refer to Section 1.1.4.3.1. 10. **JMS RESOURCE STATS** – Specify the name of the JMS destination that you have created for storing the Resource statistics. If you chose to monitor the target IIB server in an agentless manner, refer to Section 1.1.4.3.2 and if you choose to monitor the target IIB server in an agent based manner, refer to Section 1.1.3.2.2. 11. **JMS FLOW STATS** – Specify the name of the JMS destination that you have created for storing the Message flow statistics. If you chose to monitor the target IIB server in an agentless manner, refer to Section 1.1.4.3.2 and if you choose to monitor the target IIB server in an agent based manner, refer to Section 1.1.3.2.2.		
Outputs of the test	One set of results for each *Execution Group:TCP Client node* of the IBM Integration Bus that is to be monitored		
Measurements made by the	**Measurement**	**Measurement Unit**	**Interpretation**

test	**Open connections:** Indicates the number of connections that were open on this TCP Client node during the last measurement period.	Number	
	Closed connections: Indicates the number of connections that were closed on this TCP Client node since the start of the Integration server.	Number	
	Failed connections: Indicates the total number of attempted connections that failed since the start of the Integration server.	Number	Ideally, the value of this measure should be zero.
	Message received rate: Indicates the rate at which messages were received by this TCP Client node during the last measurement period.	Messages/sec	These measures are good indicators of the load on the TCP Client node. Comparing the value of these measure across the TCP client nodes will help ypu identify the node that is busy processing messages/data.
	Message sent rate: Indicates the rate at which messages were sent through this TCP Client node during the last measurement period.	Messages/sec	
	Data received rate: Indicates the rate at which data is received by this TCP Client node during the last measurement period.	Bytes/sec	
	Data sent rate: Indicates the rate at which data is sent by this TCP Client node during the last measurement period.	Bytes/sec	

2.2.6 TCP Server node statistics Test

This test monitors the TCP connections to the TCP server node and reports the count of the connections that were open, closed and failed. In the process, this test proactively alerts administrators to processing bottlenecks on the TCP server node.

Purpose	Monitors the TCP connections to the TCP server node and reports the count of the connections that were open, closed and failed. In the process, this test proactively alerts administrators to

	processing bottlenecks on the TCP server node		
Target of the test	An IBM Integration server		
Agent deploying the test	An internal/remote agent		
Configurable parameters for the test	1. **TEST PERIOD** - How often should the test be executed 2. **HOST** - The host for which the test is to be configured 3. **PORT** – The port on which the specified host listens 4. **JNDI NAMESPACE LOCATION** – Indicate where the JNDI namespace has been created. For this, select one of the following options: **File** or **LDAP**. Select **File** if the JNDI namespace is created in the same location as that of the IIB server. 5. **SSL** - If you have chosen **LDAP** as the **JNDI NAMESPACE LOCATION** wherein the LDAP server communicating with the IIB server that is to be monitored is an SSL-enabled server, then set the **SSL** flag to **Yes**. By default, this flag is set to **No**. 6. **USER DN** and **PASSWORD** – Here, provide the credentials of the Active Directory user created as explained in Section 1.1.4.1, if the target IIB server is monitored in an agentless manner. If the IIB server is to be monitored in an agent based manner, specify *none* against these parameters. 7. **CONFIRM PASSWORD** – Confirm the **PASSWORD** by retyping it in this text box. 8. **JNDI PROVIDER URL** - Specify the URL that was specified as the provider URL while creating the initial context. To know the Provider URL, refer to Figure 7 if you choose to monitor the target IIB server in an agent based manner and Figure 34 if you chose to monitor the target IIB server in an agentless manner. 9. **TOPICCONNECTIONFACTORY** – Specify the name of the connection factory in this text box. If you have chosen to monitor the target IIB server in an agentless manner, refer to Section 1.1.3.2.1 and if you have chosen to monitor the target IIB server in an agentless manner, refer to Section 1.1.4.3.1. 10. **JMS RESOURCE STATS** – Specify the name of the JMS destination that you have created for storing the Resource statistics. If you chose to monitor the target IIB server in an agentless manner, refer to Section 1.1.4.3.2 and if you choose to monitor the target IIB server in an agent based manner, refer to Section 1.1.3.2.2. 11. **JMS FLOW STATS** – Specify the name of the JMS destination that you have created for storing the Message flow statistics. If you chose to monitor the target IIB server in an agentless manner, refer to Section 1.1.4.3.2 and if you choose to monitor the target IIB server in an agent based manner, refer to Section 1.1.3.2.2.		
Outputs of the test	One set of results for each *Execution group:TCP server node* of the IBM Integration Bus that is to be monitored		
Measurements made by the	**Measurement**	**Measurement Unit**	**Interpretation**

test	**Open connections:** Indicates the number of connection that were open on this TCP server node during the last measurement period.	Number	
	Closed connections: Indicates the number of connections that were closed on this TCP server node during the last measurement period.	Number	
	Failed SSL connections: Indicates the number of attempted inbound SSL connections from external clients that failed or refused since the start of the Integration server.	Number	Ideally, the value of this measure should be zero.
	Message received rate: Indicates the rate at which messages were received by this TCP server node during the last measurement period.	Messages/sec	These measures are good indicators of the load on the TCP server node. Comparing the value of these measure across the TCP server nodes will help you identify the node that is busy processing messages/data.
	Message sent rate: Indicates the rate at which messages were sent through this TCP server node during the last measurement period.	Messages/sec	
	Data received rate: Indicates the rate at which data was received by this TCP server node during the last measurement period.	Bytes/sec	
	Data sent rate: Indicates the rate at which data was sent through this TCP server node during the last measurement period.	Bytes/sec	

2.3 The IIB Application Resources Layer

The tests associated with this layer helps you figure out the following:

- The number of successful requests and the failed requests to the CICS Transaction server;
- The number of decisions processed successfully and the decisions that failed in the Decision service;

- The garbage collection activity performed on the heap;
- The request serving capability and the utilization of the global cache;
- The processing rate of each parser type;

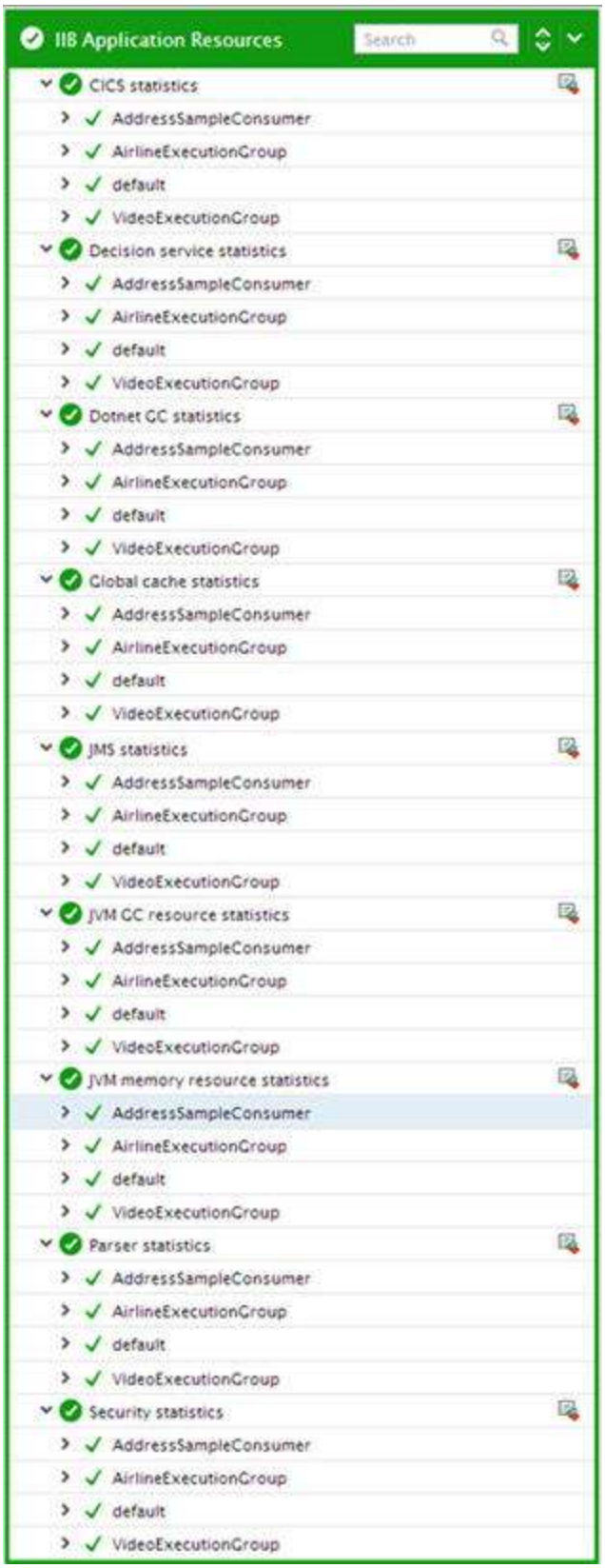
IIB Application Resources
Search
CICS statistics
AddressSampleConsumer
AirlineExecutionGroup
default
VideoExecutionGroup
Decision service statistics
AddressSampleConsumer
AirlineExecutionGroup
default
VideoExecutionGroup
Dotnet GC statistics
AddressSampleConsumer
AirlineExecutionGroup
default
VideoExecutionGroup
Global cache statistics
AddressSampleConsumer
AirlineExecutionGroup
default
VideoExecutionGroup
JMS statistics
AddressSampleConsumer
AirlineExecutionGroup
default
VideoExecutionGroup
JVM GC resource statistics
AddressSampleConsumer
AirlineExecutionGroup
default
VideoExecutionGroup
JVM memory resource statistics
AddressSampleConsumer
AirlineExecutionGroup
default
VideoExecutionGroup
Parser statistics
AddressSampleConsumer
AirlineExecutionGroup
default
VideoExecutionGroup
Security statistics
AddressSampleConsumer
AirlineExecutionGroup
default
VideoExecutionGroup

Figure 59: The tests mapped to the IIB Application Resources layer

2.3.1 CICS statistics Test

This test auto discovers the CICS Transaction servers integrated with the target IBM Integration Bus and reports the numerical statistics of the requests that were successfly and the requests that failed. Using this test administrators can identify the processing ability of the CICS Transaction server and be proactively alerted to processing bottlenecks, if any.

Purpose	Auto discovers the CICS Transaction servers integrated with the target IBM Integration Bus and reports the numerical statistics of the requests that were successfly and the requests that failed. Using this test administrators can identify the processing ability of the CICS Transaction server and be proactively alerted to processing bottlenecks, if any.
Target of the test	An IBM Integration Bus
Agent deploying the test	An internal/remote agent.

<table>
<tr><td>Configurable parameters for the test</td><td colspan="3">1. TEST PERIOD - How often should the test be executed
2. HOST - The host for which the test is to be configured
3. PORT – The port on which the specified host listens
4. JNDI NAMESPACE LOCATION – Indicate where the JNDI namespace has been created. For this, select one of the following options: File or LDAP. Select File if the JNDI namespace is created in the same location as that of the IIB server.
5. SSL - If you have chosen LDAP as the JNDI NAMESPACE LOCATION wherein the LDAP server communicating with the IIB server that is to be monitored is an SSL-enabled server, then set the SSL flag to Yes. By default, this flag is set to No.
6. USER DN and PASSWORD – Here, provide the credentials of the Active Directory user created as explained in Section 1.1.4.1, if the target IIB server is monitored in an agentless manner. If the IIB server is to be monitored in an agent based manner, specify none against these parameters.
7. CONFIRM PASSWORD – Confirm the PASSWORD by retyping it in this text box.
8. JNDI PROVIDER URL - Specify the URL that was specified as the provider URL while creating the initial context. To know the Provider URL, refer to Figure 7 if you choose to monitor the target IIB server in an agent based manner and Figure 34 if you chose to monitor the target IIB server in an agentless manner.
9. TOPICCONNECTIONFACTORY – Specify the name of the connection factory in this text box. If you have chosen to monitor the target IIB server in an agentless manner, refer to Section 1.1.3.2.1 and if you have chosen to monitor the target IIB server in an agentless manner, refer to Section 1.1.4.3.1.
10. JMS RESOURCE STATS – Specify the name of the JMS destination that you have created for storing the Resource statistics. If you chose to monitor the target IIB server in an agentless manner, refer to Section 1.1.4.3.2 and if you choose to monitor the target IIB server in an agent based manner, refer to Section 1.1.3.2.2.
11. JMS FLOW STATS – Specify the name of the JMS destination that you have created for storing the Message flow statistics. If you chose to monitor the target IIB server in an agentless manner, refer to Section 1.1.4.3.2 and if you choose to monitor the target IIB server in an agent based manner, refer to Section 1.1.3.2.2.</td></tr>
<tr><td>Outputs of the test</td><td colspan="3">One set of results for each Execution Group:Z/OS region:CICS Transaction server of the IBM Integration Bus that is to be monitored</td></tr>
<tr><td rowspan="2">Measurements made by the test</td><td>Measurement</td><td>Measurement Unit</td><td>Interpretation</td></tr>
<tr><td>Successful requests:
Indicates the number of requests from the Integration node to this CICS Transaction server that were successful during the last measurement period.</td><td>Number</td><td>A high value is desired for this measure. If the value of this measure is decreasing alarmingly, then it may indicate a performance bottleneck/slowdown of the server/network.</td></tr>
</table>

	Request failures: Indicates the total number of requests to this CICS Transaction server that failed during the last measurement period.	Number	The value of this measure does not include the value of the *Connection attempt failures* measure.
	Requests failed due to security validations: Indicates the number of requests to this CICS Transaction server that failed due to security validation during the last measurement period.	Number	The requests may fail due to security issues such as authentication failure, improper port configuration etc.
	Connection attempt failures: Indicates the number of connections to this CICS Transaction server that failed during the last measurement period.	Number	

2.3.2 Decision service statistics Test

Most enterprise applications today are very dynamic in nature. Customer scenarios often require querying database tables for fetching values, which are then used to execute the rules instead of using static data.

IBM Operational Decision Manager (ODM) is IBM's next generation Business Rule Management System (BRMS). It is a full-featured, easy-to-use platform for capturing, automating and governing frequent, repeatable *business decisions* that drive critical business processes, applications and systems. In IBM ODM, business decisions are exposed as decision services. Decision services are reusable service operations. The decision service isolates the logic behind business decisions, separating it from business processes and the application code.

Although good practices advise that a business rule application should not access external data, there are situations when rule applications need to access external data to make decisions. One example is when a rule application needs dynamic data instead of static data is a database table. A rule application might be needed to look up a database table to fetch values, which are then used to run the business rules. Administrators may want to know how well the business rules are run in their envornment and how many rules have been successful. The **Decision service statistics** test helps them achieve the same! Using this test, administrators may figure out the number of decisions that were processed successfully and the decisions that failed. Additionally, this test helps administrators figure out the rules that were triggered by the messages processed by the decision service.

Purpose	Using this test, administrators may figure out the number of decisions that were processed successfully and the decisions that failed. Additionally, this test helps administrators figure out the rules that were triggered by the messages processed by the decision service.
Target of the test	An IBM Integration Bus

Agent deploying the test	An internal/remote agent		
Configurable parameters for the test	1. **TEST PERIOD** - How often should the test be executed 2. **HOST** - The host for which the test is to be configured 3. **PORT** – The port on which the specified host listens 4. **JNDI NAMESPACE LOCATION** – Indicate where the JNDI namespace has been created. For this, select one of the following options: **File** or **LDAP**. Select **File** if the JNDI namespace is created in the same location as that of the IIB server. 5. **SSL** - If you have chosen **LDAP** as the **JNDI NAMESPACE LOCATION** wherein the LDAP server communicating with the IIB server that is to be monitored is an SSL-enabled server, then set the **SSL** flag to **Yes**. By default, this flag is set to **No**. 6. **USER DN** and **PASSWORD** – Here, provide the credentials of the Active Directory user created as explained in Section 1.1.4.1, if the target IIB server is monitored in an agentless manner. If the IIB server is to be monitored in an agent based manner, specify *none* against these parameters. 7. **CONFIRM PASSWORD** – Confirm the **PASSWORD** by retyping it in this text box. 8. **JNDI PROVIDER URL** - Specify the URL that was specified as the provider URL while creating the initial context. To know the Provider URL, refer to Figure 7 if you choose to monitor the target IIB server in an agent based manner and Figure 34 if you chose to monitor the target IIB server in an agentless manner. 9. **TOPICCONNECTIONFACTORY** – Specify the name of the connection factory in this text box. If you have chosen to monitor the target IIB server in an agentless manner, refer to Section 1.1.3.2.1 and if you have chosen to monitor the target IIB server in an agentless manner, refer to Section 1.1.4.3.1. 10. **JMS RESOURCE STATS** – Specify the name of the JMS destination that you have created for storing the Resource statistics. If you chose to monitor the target IIB server in an agentless manner, refer to Section 1.1.4.3.2 and if you choose to monitor the target IIB server in an agent based manner, refer to Section 1.1.3.2.2. 11. **JMS FLOW STATS** – Specify the name of the JMS destination that you have created for storing the Message flow statistics. If you chose to monitor the target IIB server in an agentless manner, refer to Section 1.1.4.3.2 and if you choose to monitor the target IIB server in an agent based manner, refer to Section 1.1.3.2.2.		
Outputs of the test	One set of results for each *Execution group:decision service* that is to be monitored		
Measurements made by the	**Measurement**	**Measurement Unit**	**Interpretation**

test	**Successful decisions:** Indicates the number of decisions that were processed successfully by this decision service during the last measurement period.	Number	A high value is desired for this measure.
	Failed decisions: Indicates the number of decisions that were not processed successfully by this decision service i.e., the decisions that failed during the last measurement period.	Number	Ideally, the value of this measure should be zero. An sudden/gradual increase in the value is a cause of concern which requires the immediate attention of the administrators.
	Rules matched: Indicates the total number of rules that were triggered (matched) by the messages processed by this decision service during the last measurement period.	Number	

2.3.3 Dotnet GC statistics Test

This test monitors the memory allocation activity of each ASP .Net server integrated with the IBM Integration Bus, in terms of heaps when objects are created and managed.

This test will report metrics for WebSphere Message Broker 8 and above only.

Purpose	Monitors the memory allocation activity of each ASP .Net server integrated with the IBM Integration Bus, in terms of heaps when objects are created and managed
Target of the test	An IBM Integration Bus
Agent deploying the test	An internal/remote agent

<table>
<tr><td>Configurable parameters for the test</td><td colspan="3">1. TEST PERIOD - How often should the test be executed
2. HOST - The host for which the test is to be configured
3. PORT – The port on which the specified host listens
4. JNDI NAMESPACE LOCATION – Indicate where the JNDI namespace has been created. For this, select one of the following options: File or LDAP. Select File if the JNDI namespace is created in the same location as that of the IIB server.
5. SSL - If you have chosen LDAP as the JNDI NAMESPACE LOCATION wherein the LDAP server communicating with the IIB server that is to be monitored is an SSL-enabled server, then set the SSL flag to Yes. By default, this flag is set to No.
6. USER DN and PASSWORD – Here, provide the credentials of the Active Directory user created as explained in Section 1.1.4.1, if the target IIB server is monitored in an agentless manner. If the IIB server is to be monitored in an agent based manner, specify none against these parameters.
7. CONFIRM PASSWORD – Confirm the PASSWORD by retyping it in this text box.
8. JNDI PROVIDER URL - Specify the URL that was specified as the provider URL while creating the initial context. To know the Provider URL, refer to Figure 7 if you choose to monitor the target IIB server in an agent based manner and Figure 34 if you chose to monitor the target IIB server in an agentless manner.
9. TOPICCONNECTIONFACTORY – Specify the name of the connection factory in this text box. If you have chosen to monitor the target IIB server in an agentless manner, refer to Section 1.1.3.2.1 and if you have chosen to monitor the target IIB server in an agentless manner, refer to Section 1.1.4.3.1.
10. JMS RESOURCE STATS – Specify the name of the JMS destination that you have created for storing the Resource statistics. If you chose to monitor the target IIB server in an agentless manner, refer to Section 1.1.4.3.2 and if you choose to monitor the target IIB server in an agent based manner, refer to Section 1.1.3.2.2.
11. JMS FLOW STATS – Specify the name of the JMS destination that you have created for storing the Message flow statistics. If you chose to monitor the target IIB server in an agentless manner, refer to Section 1.1.4.3.2 and if you choose to monitor the target IIB server in an agent based manner, refer to Section 1.1.3.2.2.</td></tr>
<tr><td>Outputs of the test</td><td colspan="3">One set of results for the Execution group:.Net server of the IBM Integration Bus that is to be monitored</td></tr>
<tr><td rowspan="2">Measurements made by the test</td><td>Measurement</td><td>Measurement Unit</td><td>Interpretation</td></tr>
<tr><td>Explicit GC count:
Indicates the number of garbage collections that were forced by an external request for this server during the last measurement period.</td><td>Number</td><td></td></tr>
</table>

	Generation 0 collections taken: Indicates the number of generation 0 objects (youngest; most recently allocated) that were garbage collected (Gen 0 GC) since the start of the application.	Number	
	Generation 1 collections taken: Indicates the number of generation 1 objects that have been garbage collected since the start of the application.	Number	Objects that survive are promoted to generation 2.
	Generation 2 collections taken: Indicates the number of generation 2 objects that have been garbage collected since the start of the application.	Number	Generation 2 is the highest, thus objects that survive collection remain in generation 2. Generation 2 collections can be very expensive, especially if the size of the Generation 2 heap is huge.
	Reserved in all heaps: Indicates the amount of memory in bytes that are reserved in all heaps.	MB	
	Generation zero heap size: Indicates the maximum amount of bytes that can be allocated in generation 0.	MB	A generation 0 garbage collection occurs when the allocations since the last collection exceed this size. The generation 0 size is tuned by the garbage collector and can change during the execution of the application. At the end of a generation 0 collection the size of the generation 0 heap is 0 bytes. This measure displays the size, in bytes, of allocations that invokes the next generation 0 garbage collection. This measure is updated at the end of a garbage collection, not at each allocation.
	Generation one heap size: Indicates the amount of bytes in generation 1.	MB	Objects are not directly allocated in this generation; they are promoted from previous generation 0 garbage collections. This measure is updated at the end of a garbage collection, not at each allocation.

	Generation two heap size: Indicates the amount of bytes in generation 2.	MB	Objects are not directly allocated in this generation; they are promoted from generation 1 during previous generation 1 garbage collections. This measure is updated at the end of a garbage collection, not at each allocation.
	Largest object heap size: Indicates the current size of the largest object heap.	MB	Objects that are greater than approximately 85,000 bytes are treated as large objects by the garbage collector and are directly allocated in a special heap; they are not promoted through the generations. This counter is updated at the end of a garbage collection, not at each allocation.
	Promoted from generation 0 to 1: Indicates the amount of memory that survived garbage collection and are promoted from generation 0 to generation 1.	MB	Objects that are promoted only because they are waiting to be finalized are not included in this measure. This measure displays the value observed at the end of the last garbage collection.
	Promoted from generation 1 to 2: Indicates the amount of memory that survived garbage collection and are promoted from generation 1 to generation 2.	MB	Objects that are promoted only because they are waiting to be finalized are not included in this measure. This measure displays the value observed at the end of the last garbage collection. This measure is reset to 0 if the last garbage collection was a generation 0 collection only.

2.3.4 Global cache statistics Test

The global cache is embedded in the integration node of the IBM Integration Bus. This global cache is a repository for data that you want to reuse. For example, you can use a global cache in WebSphere MQ message flows to store correlation information for use beyond a specific message flow node, instance of a message flow, integration server, or integration node. The cache facilitates sharing of data across processes (both in the same integration node, and across integration nodes) and eliminates the need for an alternative solution, such as a database. You can use one message flow node to store data in the global cache, then a second node (in the same message flow or a separate flow), can retrieve that data from the global cache.

You can use a message flow node to interact with the global cache. Interactions with the cache happen outside the message flow transaction, and are committed immediately. If an exception is thrown downstream of the node that interacts with the cache, the cache interactions are not rolled back.

A global cache is said to be effectively utilized only if it is able to service the maximum number of requests to the IBM Integration Bus; this greatly reduces direct data accesses and related overheads, and thus improving the server performance. On the contrary, ineffective cache usage can be the key contributor to a slowdown or degradation in server performance, as it increases direct data accesses.

This test monitors each global cache of the IBM Integration Bus and reports its usage - in terms of its request serving ability. In the process, the test proactively alerts administrators to the under-utilization of the global cache and the failures in connecting to the cache, and helps them quickly initiate corrective measures.

Purpose	Monitors each global cache of the IBM Integration Bus and reports its usage - in terms of its request serving ability. In the process, the test proactively alerts administrators to the under-utilization of the global cache and the failures in connecting to the cache, and helps them quickly initiate corrective measures
Target of the test	An IBM Integration Bus
Agent deploying the test	An internal/remote agent

<table>
<tr><td>Configurable parameters for the test</td><td colspan="3">1. TEST PERIOD - How often should the test be executed
2. HOST - The host for which the test is to be configured
3. PORT – The port on which the specified host listens
4. JNDI NAMESPACE LOCATION – Indicate where the JNDI namespace has been created. For this, select one of the following options: File or LDAP. Select File if the JNDI namespace is created in the same location as that of the IIB server.
5. SSL - If you have chosen LDAP as the JNDI NAMESPACE LOCATION wherein the LDAP server communicating with the IIB server that is to be monitored is an SSL-enabled server, then set the SSL flag to Yes. By default, this flag is set to No.
6. USER DN and PASSWORD – Here, provide the credentials of the Active Directory user created as explained in Section 1.1.4.1, if the target IIB server is monitored in an agentless manner. If the IIB server is to be monitored in an agent based manner, specify none against these parameters.
7. CONFIRM PASSWORD – Confirm the PASSWORD by retyping it in this text box.
8. JNDI PROVIDER URL - Specify the URL that was specified as the provider URL while creating the initial context. To know the Provider URL, refer to Figure 7 if you choose to monitor the target IIB server in an agent based manner and Figure 34 if you chose to monitor the target IIB server in an agentless manner.
9. TOPICCONNECTIONFACTORY – Specify the name of the connection factory in this text box. If you have chosen to monitor the target IIB server in an agentless manner, refer to Section 1.1.3.2.1 and if you have chosen to monitor the target IIB server in an agentless manner, refer to Section 1.1.4.3.1.
10. JMS RESOURCE STATS – Specify the name of the JMS destination that you have created for storing the Resource statistics. If you chose to monitor the target IIB server in an agentless manner, refer to Section 1.1.4.3.2 and if you choose to monitor the target IIB server in an agent based manner, refer to Section 1.1.3.2.2.
11. JMS FLOW STATS – Specify the name of the JMS destination that you have created for storing the Message flow statistics. If you chose to monitor the target IIB server in an agentless manner, refer to Section 1.1.4.3.2 and if you choose to monitor the target IIB server in an agent based manner, refer to Section 1.1.3.2.2.</td></tr>
<tr><td>Outputs of the test</td><td colspan="3">One set of results for each Execution group:global cache/configurable service of the IBM Integration Bus that is to be monitored</td></tr>
<tr><td rowspan="2">Measurements made by the test</td><td>Measurement</td><td>Measurement Unit</td><td>Interpretation</td></tr>
<tr><td>Successful connects to the cache:
Indicates the number of successful attempts that were made from the integration server to this global cache during the last measurement period.</td><td>Number</td><td></td></tr>
</table>

	Map reads: Indicates the number of read operations that were completed by message flows in the integration server on this global cache during the last measurement period.	Number	A high value is desired for this measure. A low value or a consistent drop in this value is a cause for concern, as it indicates ineffective cache usage. This means high direct accesses of data, and poor server performance. One of the common reasons for bad cache usage is improper cache size. A cache that does not have sufficient space to accommodate entries will not be able to service requests effectively. You may hence have to allocate more space to the cache or free-up space in the cache to accommodate more entries.
	Map writes: Indicates the number of write operations that were completed by message flows in the integration server on this global cache during the last measurement period.	Number	This measure is incremented when adding or updating an entry in the cache. A high value is desired for this measure. A low value or a consistent drop in this value is a cause for concern, as it indicates ineffective cache usage. This means high disk accesses, and poor server performance. One of the common reasons for bad cache usage is improper cache size. A cache that does not have sufficient space to accommodate entries will not be able to service requests effectively. You may hence have to allocate more space to the cache or free-up space in the cache to accommodate more entries.
	Map removes: Indicates the number of remove operations that were completed by message flows in the integration server on this global cache during the last measurement period.	Number	This measure is incremented when an entry is removed/updated from the cache.
	Message failed operations on the cache map: Indicates the number of failed map operations by message flows in the integration server on this global cache during the last measurement period.	Number	
	Map used: Indicates the total number of maps used by message flows in the integration server on this global cache during the last measurement period.	Number	

	Connection failures: Indicates the number of failed attempts to connect from the integration server to the global cache during the last measurement period.	Number	A low value is desired for this measure. A high value or a consistent increase in this value is a cause for concern, as it indicates ineffective cache usage.
	Total map actions: Indicates the number of map operations that were completed by message flows in the integration server on the global cache during the last measurement period.	Number	The value of this measure includes reads, writes, removes, and key checks.

2.3.5 JMS statistics Test

This test reports the performance statistics of each JMS Connection factory in the IBM Integration Bus. Using this test, administrators can be proactively alerted to potential processing bottlenecks in the JMS Connection factory.

Purpose	Reports the performance statistics of each JMS Connection factory in the IBM Integration Bus
Target of the test	An IBM integration Bus
Agent deploying the test	An interna/remote agent

<table>
<tr><td>Configurable parameters for the test</td><td colspan="3">1. TEST PERIOD - How often should the test be executed
2. HOST - The host for which the test is to be configured
3. PORT – The port on which the specified host listens
4. JNDI NAMESPACE LOCATION – Indicate where the JNDI namespace has been created. For this, select one of the following options: File or LDAP. Select File if the JNDI namespace is created in the same location as that of the IIB server.
5. SSL - If you have chosen LDAP as the JNDI NAMESPACE LOCATION wherein the LDAP server communicating with the IIB server that is to be monitored is an SSL-enabled server, then set the SSL flag to Yes. By default, this flag is set to No.
6. USER DN and PASSWORD – Here, provide the credentials of the Active Directory user created as explained in Section 1.1.4.1, if the target IIB server is monitored in an agentless manner. If the IIB server is to be monitored in an agent based manner, specify none against these parameters.
7. CONFIRM PASSWORD – Confirm the PASSWORD by retyping it in this text box.
8. JNDI PROVIDER URL - Specify the URL that was specified as the provider URL while creating the initial context. To know the Provider URL, refer to Figure 7 if you choose to monitor the target IIB server in an agent based manner and Figure 34 if you chose to monitor the target IIB server in an agentless manner.
9. TOPICCONNECTIONFACTORY – Specify the name of the connection factory in this text box. If you have chosen to monitor the target IIB server in an agentless manner, refer to Section 1.1.3.2.1 and if you have chosen to monitor the target IIB server in an agentless manner, refer to Section 1.1.4.3.1.
10. JMS RESOURCE STATS – Specify the name of the JMS destination that you have created for storing the Resource statistics. If you chose to monitor the target IIB server in an agentless manner, refer to Section 1.1.4.3.2 and if you choose to monitor the target IIB server in an agent based manner, refer to Section 1.1.3.2.2.
11. JMS FLOW STATS – Specify the name of the JMS destination that you have created for storing the Message flow statistics. If you chose to monitor the target IIB server in an agentless manner, refer to Section 1.1.4.3.2 and if you choose to monitor the target IIB server in an agent based manner, refer to Section 1.1.3.2.2.</td></tr>
<tr><td>Outputs of the test</td><td colspan="3">One set of results for each Execution Group:JMS Connection factory_JNDI bindings location of the IBM Integration Bus that is to be monitored</td></tr>
<tr><td rowspan="2">Measurements made by the test</td><td>Measurement</td><td>Measurement Unit</td><td>Interpretation</td></tr>
<tr><td>Open JMS connections:
Indicates the number of JMS connections that are currently open in this JMS Connection factory.</td><td>Number</td><td></td></tr>
</table>

	Open JMS sessions: Indicates the number of JMS sessions that are currently open in this JMS Connection factory.	Number	
	Message received rate: Indicates the rate at which messages were received byJMSInput or JMSReceive nodes in this JMS Connection factory during the last measurement period.	Messages/sec	Comparing the value of these measures across the JMS Connection factories will help you identify the JMS Connection factory that is most busy receiving/sending messages.
	Message sent rate: Indicates the rate at which messages are sent by JMSOutput nodes in this JMS Connection factory during the last measurement period.	Messages/sec	
	Message browsed rate: Indicates the rate at which messages were received by the JMSReceive nodes in this JMS Connection factory during the last measurement period.	Messages/sec	
	JMS connection failures: Indicates the total number of attempted JMS connections that failed since the time the integration server was last restarted during the last measurement period.	Number	Ideally, the value of this measure should be zero.

2.3.6 JVM GC resource statistics Test

Manual memory management is time consuming, and error prone. Most programs still contain leaks. This is all doubly true with programs using exception-handling and/or threads. Garbage collection (GC) is a part of a Java application's JVM that automatically determines what memory a program is no longer using, and recycles it for other use. It is also known as "automatic storage (or memory) reclamation". The **JVM GC resource statistics** test reports the performance statistics pertaining to the JVM's garbage collection in the target IBM Integration Bus.

Purpose	Reports the performance statistics pertaining to the JVM's garbage collection in the target IBM Integration Bus
Target of the test	An IBM Integration Bus
Agent deploying the	An internal/remote agent

<table>
<tr><td>test</td><td colspan="3"></td></tr>
<tr><td>Configurable parameters for the test</td><td colspan="3">1. TEST PERIOD - How often should the test be executed
2. HOST - The host for which the test is to be configured
3. PORT – The port on which the specified host listens
4. JNDI NAMESPACE LOCATION – Indicate where the JNDI namespace has been created. For this, select one of the following options: File or LDAP. Select File if the JNDI namespace is created in the same location as that of the IIB server.
5. SSL - If you have chosen LDAP as the JNDI NAMESPACE LOCATION wherein the LDAP server communicating with the IIB server that is to be monitored is an SSL-enabled server, then set the SSL flag to Yes. By default, this flag is set to No.
6. USER DN and PASSWORD – Here, provide the credentials of the Active Directory user created as explained in Section 1.1.4.1, if the target IIB server is monitored in an agentless manner. If the IIB server is to be monitored in an agent based manner, specify none against these parameters.
7. CONFIRM PASSWORD – Confirm the PASSWORD by retyping it in this text box.
8. JNDI PROVIDER URL - Specify the URL that was specified as the provider URL while creating the initial context. To know the Provider URL, refer to Figure 7 if you choose to monitor the target IIB server in an agent based manner and Figure 34 if you chose to monitor the target IIB server in an agentless manner.
9. TOPICCONNECTIONFACTORY – Specify the name of the connection factory in this text box. If you have chosen to monitor the target IIB server in an agentless manner, refer to Section 1.1.3.2.1 and if you have chosen to monitor the target IIB server in an agentless manner, refer to Section 1.1.4.3.1.
10. JMS RESOURCE STATS – Specify the name of the JMS destination that you have created for storing the Resource statistics. If you chose to monitor the target IIB server in an agentless manner, refer to Section 1.1.4.3.2 and if you choose to monitor the target IIB server in an agent based manner, refer to Section 1.1.3.2.2.
11. JMS FLOW STATS – Specify the name of the JMS destination that you have created for storing the Message flow statistics. If you chose to monitor the target IIB server in an agentless manner, refer to Section 1.1.4.3.2 and if you choose to monitor the target IIB server in an agent based manner, refer to Section 1.1.3.2.2.</td></tr>
<tr><td>Outputs of the test</td><td colspan="3">One set of results for each Execution group:garbage collector that is reclaiming the unused memory on the JVM of the server being monitored</td></tr>
<tr><td rowspan="2">Measurements made by the test</td><td>Measurement</td><td>Measurement Unit</td><td>Interpretation</td></tr>
<tr><td>Number Of GC collections:
Indicates the number of times this garbage collector was started to release dead objects from memory during the last measurement period.</td><td>Number</td><td></td></tr>
</table>

	GC duration: Indicates the time taken to by this garbage collector to perform the current garbage collection operation.	Secs	Ideally, the value of both these measures should be low. This is because, the garbage collection (GC) activity tends to suspend the operations of the application until such time that GC ends. Longer the GC time, longer it would take for the application to resume its functions. To minimize the impact of GC on application performance, it is best to ensure that GC activity does not take too long to complete.
	GC duration as percentage over test frequency: Indicates the percentage of time spent by this garbage collector on garbage collection during the last measurement period.	Percent	

2.3.7 JVM memory resource statistics Test

This test monitors every memory type on the JVM of the target IBM Integration Bus and reports how efficiently the JVM utilizes the memory resources of each type.

Purpose	Monitors every memory type on the JVM of the target IBM Integration Bus and reports how efficiently the JVM utilizes the memory resources of each type.
Target of the test	An IBM Integration Bus
Agent deploying the test	An internal/remote agent

<table>
<tr><td>Configurable parameters for the test</td><td colspan="3">1. TEST PERIOD - How often should the test be executed
2. HOST - The host for which the test is to be configured
3. PORT – The port on which the specified host listens
4. JNDI NAMESPACE LOCATION – Indicate where the JNDI namespace has been created. For this, select one of the following options: File or LDAP. Select File if the JNDI namespace is created in the same location as that of the IIB server.
5. SSL - If you have chosen LDAP as the JNDI NAMESPACE LOCATION wherein the LDAP server communicating with the IIB server that is to be monitored is an SSL-enabled server, then set the SSL flag to Yes. By default, this flag is set to No.
6. USER DN and PASSWORD – Here, provide the credentials of the Active Directory user created as explained in Section 1.1.4.1, if the target IIB server is monitored in an agentless manner. If the IIB server is to be monitored in an agent based manner, specify none against these parameters.
7. CONFIRM PASSWORD – Confirm the PASSWORD by retyping it in this text box.
8. JNDI PROVIDER URL - Specify the URL that was specified as the provider URL while creating the initial context. To know the Provider URL, refer to Figure 7 if you choose to monitor the target IIB server in an agent based manner and Figure 34 if you chose to monitor the target IIB server in an agentless manner.
9. TOPICCONNECTIONFACTORY – Specify the name of the connection factory in this text box. If you have chosen to monitor the target IIB server in an agentless manner, refer to Section 1.1.3.2.1 and if you have chosen to monitor the target IIB server in an agentless manner, refer to Section 1.1.4.3.1.
10. JMS RESOURCE STATS – Specify the name of the JMS destination that you have created for storing the Resource statistics. If you chose to monitor the target IIB server in an agentless manner, refer to Section 1.1.4.3.2 and if you choose to monitor the target IIB server in an agent based manner, refer to Section 1.1.3.2.2.
11. JMS FLOW STATS – Specify the name of the JMS destination that you have created for storing the Message flow statistics. If you chose to monitor the target IIB server in an agentless manner, refer to Section 1.1.4.3.2 and if you choose to monitor the target IIB server in an agent based manner, refer to Section 1.1.3.2.2.</td></tr>
<tr><td>Outputs of the test</td><td colspan="3">One set of results for each memory type on the JVM of the IBM Integration Bus that is to be monitored</td></tr>
<tr><td rowspan="2">Measurements made by the test</td><td>Measurement</td><td>Measurement Unit</td><td>Interpretation</td></tr>
<tr><td>Committed memory:
Indicates the amount of memory that is allocated to this memory type on the JVM by the operating system.</td><td>MB</td><td></td></tr>
</table>

	Used memory: Indicates the amount of memory of this memory type that is currently in use.	MB	It includes the memory occupied by all objects, including both reachable and unreachable objects. Ideally, the value of this measure should be low. A high value or a consistent increase in the value could indicate gradual erosion of memory resources. In such a situation, you can take the help of the detailed diagnosis of this measure (if enabled), to figure out which class is using up memory excessively.
	Maximum configured memory: Indicates the maximum amount of memory of this memory type that can be used for memory management.	MB	
	Used percentage of committed memory: Indicates the percentage of memory that is allocated to this memory type on the JVM.	Percent	Ideally, the value of this measure should be low. A very high value of this measure could indicate excessive memory consumption by the JVM, which in turn, could warrant further investigation.
	Used percentage of the maximum configured memory: Indicates the percentage of maximum memory of this memory type that can be used for memory management.	Percent	
	Free memory in the committed memory: Indicates the amount of memory that is left unused from the total amount allocated to this memory type on the JVM.	MB	A high value is desired for this measure.
	Free memory in the maximum configured memory: Indicates the amount of memory that is unused from the maximum amount of memory of this memory type that can be used for memory management.	MB	A high value is desired for this measure.

2.3.8 Parser statistics Test

A parser is a program that interprets the physical bit stream of an incoming message, and creates an internal logical representation of the message in a tree structure. The parser also regenerates a bit stream for an outgoing message from the internal message tree representation. A parser is called when the bit stream that represents an input message is converted to the internal form that can be handled by the broker; this invocation of the parser is known as *parsing*. The internal form, a logical tree structure, is described in Logical tree structure. It is described as a tree because messages are typically hierarchical in structure; a good example of this structure is XML. The way in which the parser interprets the bit stream is unique to that parser; therefore, the logical message tree that is created from the bit stream varies from parser to parser.

The parser that is called depends on the structure of a message, referred to as the *message template*. Message template information comprises the *message domain*, *message set*, *message type*, and *physical format* of the message. Together, these values identify the structure of the data that the message contains.

A parser is also called when a logical tree that represents an output message is converted into a bit stream; this action by the parser is known as *writing*. Typically, an output message is generated by an output node at the end of the message flow. However, you can connect more nodes to an output node to continue processing of the message.

The message domain identifies the parser that is used to parse and write instances of the message. The remaining parts of the message template, message set, message type, and physical format, are optional, and are used by model-driven parsers such as the MRM parser.

For each message flow parser type, this test reports the largest bit stream that is parsed/written. In addition, this test reports the processing rate of each parser type in terms of parses and writes. This way, administrators may be alerted to processing bottlenecks , if any.

Purpose	Monitors the current memory utilization of this firewall device
Target of the test	An IBM Integration Bus
Agent deploying the test	An internal/remote agent

<table>
<tr><td>Configurable parameters for the test</td><td colspan="3">1. TEST PERIOD - How often should the test be executed
2. HOST - The host for which the test is to be configured
3. PORT – The port on which the specified host listens
4. JNDI NAMESPACE LOCATION – Indicate where the JNDI namespace has been created. For this, select one of the following options: File or LDAP. Select File if the JNDI namespace is created in the same location as that of the IIB server.
5. SSL - If you have chosen LDAP as the JNDI NAMESPACE LOCATION wherein the LDAP server communicating with the IIB server that is to be monitored is an SSL-enabled server, then set the SSL flag to Yes. By default, this flag is set to No.
6. USER DN and PASSWORD – Here, provide the credentials of the Active Directory user created as explained in Section 1.1.4.1, if the target IIB server is monitored in an agentless manner. If the IIB server is to be monitored in an agent based manner, specify none against these parameters.
7. CONFIRM PASSWORD – Confirm the PASSWORD by retyping it in this text box.
8. JNDI PROVIDER URL - Specify the URL that was specified as the provider URL while creating the initial context. To know the Provider URL, refer to Figure 7 if you choose to monitor the target IIB server in an agent based manner and Figure 34 if you chose to monitor the target IIB server in an agentless manner.
9. TOPICCONNECTIONFACTORY – Specify the name of the connection factory in this text box. If you have chosen to monitor the target IIB server in an agentless manner, refer to Section 1.1.3.2.1 and if you have chosen to monitor the target IIB server in an agentless manner, refer to Section 1.1.4.3.1.
10. JMS RESOURCE STATS – Specify the name of the JMS destination that you have created for storing the Resource statistics. If you chose to monitor the target IIB server in an agentless manner, refer to Section 1.1.4.3.2 and if you choose to monitor the target IIB server in an agent based manner, refer to Section 1.1.3.2.2.
11. JMS FLOW STATS – Specify the name of the JMS destination that you have created for storing the Message flow statistics. If you chose to monitor the target IIB server in an agentless manner, refer to Section 1.1.4.3.2 and if you choose to monitor the target IIB server in an agent based manner, refer to Section 1.1.3.2.2.</td></tr>
<tr><td>Outputs of the test</td><td colspan="3">One set of results for each Execution group:message flow parser of the IBM Integration Bus that is to be monitored</td></tr>
<tr><td rowspan="2">Measurements made by the test</td><td>Measurement</td><td>Measurement Unit</td><td>Interpretation</td></tr>
<tr><td>Threads:
Indicates the number of message flow threads that contributed to the statistics of this message flow parser type accumulation during the last measurement period.</td><td>Number</td><td></td></tr>
</table>

	Memory utilization: Indicates the approximate amount of user data-related memory used for this message flow parser type during the last measurement period.	KB	The value of this measure cannot be calculated exactly.
	Maximum read bit stream: Indicates the largest bit stream parsed by this message flow parser type during the last measurement period.	KB	
	Maximum written bit stream: Indicates the largest bit stream written by this message flow parser type during the last measurement period.	KB	
	Reads: Indicates the rate at which parses were completed successfully by this message flow parser type during the last measurement period.	Reads/sec	Comapring the value of this measure across the parser types will help you identify the parser type that is busy processing the parses.
	Failed reads: Indicates the rate at which parses failed in this message flow parser type during the last measurement period.	Reads/sec	A low value is desired for this measure. A sudden/gradual increase in the value of this measure indicates processing bottlenecks.
	Writes: Indicates the rate at which writes were completed successfully to this message flow parser type during the last measurement period.	Writes/sec	Comparing the value of this measure across the parser types will help you identify the parser type that is busy writing the messages.
	Failed writes: Indicates the rate at which parses failed to be written to this message flow parser type during the last measurement period.	Writes/sec	A low value is desired for this measure. A sudden/gradual increase in the value of this measure indicates processing bottlenecks.

2.3.9 Security statistics Test

When a message flow is configured with a security profile, requests are typically made to a security provider or security token server (STS) to process and approve authentication, mapping, or authorization. Use the **Security statistics** test to review the number of requests that are made, how many of those requests are successful, and how many are being serviced from the security cache.

Purpose	Helps you to review the number of requests that are made, how many of those requests are successful, and how many are being serviced from the security cache
Target of the test	An IBM Integration Bus
Agent deploying the test	An internal/remote agent
Configurable parameters for the test	1. **TEST PERIOD** - How often should the test be executed 2. **HOST** - The host for which the test is to be configured 3. **PORT** – The port on which the specified host listens 4. **JNDI NAMESPACE LOCATION** – Indicate where the JNDI namespace has been created. For this, select one of the following options: **File** or **LDAP**. Select **File** if the JNDI namespace is created in the same location as that of the IIB server. 5. **SSL** - If you have chosen **LDAP** as the **JNDI NAMESPACE LOCATION** wherein the LDAP server communicating with the IIB server that is to be monitored is an SSL-enabled server, then set the **SSL** flag to **Yes**. By default, this flag is set to **No**. 6. **USER DN** and **PASSWORD** – Here, provide the credentials of the Active Directory user created as explained in Section 1.1.4.1, if the target IIB server is monitored in an agentless manner. If the IIB server is to be monitored in an agent based manner, specify *none* against these parameters. 7. **CONFIRM PASSWORD** – Confirm the **PASSWORD** by retyping it in this text box. 8. **JNDI PROVIDER URL** - Specify the URL that was specified as the provider URL while creating the initial context. To know the Provider URL, refer to Figure 7 if you choose to monitor the target IIB server in an agent based manner and Figure 34 if you chose to monitor the target IIB server in an agentless manner. 9. **TOPICCONNECTIONFACTORY** – Specify the name of the connection factory in this text box. If you have chosen to monitor the target IIB server in an agentless manner, refer to Section 1.1.3.2.1 and if you have chosen to monitor the target IIB server in an agentless manner, refer to Section 1.1.4.3.1. 10. **JMS RESOURCE STATS** – Specify the name of the JMS destination that you have created for storing the Resource statistics. If you chose to monitor the target IIB server in an agentless manner, refer to Section 1.1.4.3.2 and if you choose to monitor the target IIB server in an agent based manner, refer to Section 1.1.3.2.2. 11. **JMS FLOW STATS** – Specify the name of the JMS destination that you have created for storing the Message flow statistics. If you chose to monitor the target IIB server in an agentless manner, refer to Section 1.1.4.3.2 and if you choose to monitor the target IIB server in an agent based manner, refer to Section 1.1.3.2.2.

Outputs of the test	One set of results for each *Execution group:security proivider* of the Integration server that is to be monitored		
Measurements made by the test	**Measurement**	**Measurement Unit**	**Interpretation**
	Total cache entries: Indicates the total number of security operation result entries in the security cache of this security provider.	Number	A security operation is defined in the security profile as authentication, mapping, or authorization. A cache entry might include a returned security token.
	Total security operations: Indicates the total number of security operations during the last measurement period.	Number	A security operation is defined in the security profile as authentication, mapping, or authorization. A security profile with both authentication and authorization counts as two operations.
	Successful security operations: Indicates the number of security operations that were approved during the last measurement period.	Number	
	Operations serviced by cache: Indicates the number of security operations that were serviced from the security cache during the last measurement period.	Number	A high value is desired for this measure.
	Unsuccessful security operations: Indicates the number of security operations that failed to be approved during the last measurement period.	Number	Ideally, the value of this measure should be zero.

2.4 The IIB Message Flows Layer

This layer tracks the message processing capability of each message flow node and the message processing capability of each message flow. Using the metrics collected from this layer, administrators can pinpoint the exact node that is slow in processing the messages and rectify the same instantly. Additionally, the thread that is slow in processing the messges can also be figured out using a host of metrics collected for each thread that is executing on the execution group of the IIB server.

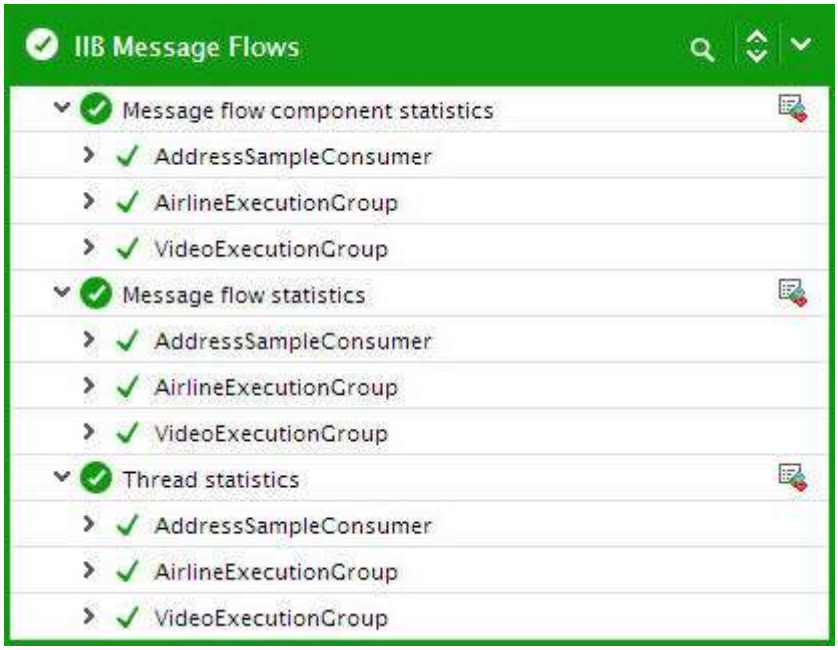

Figure 60: The tests mapped to the IIB Message Flows layer

2.4.1 Message flow component statistics Test

A message flow node is a processing step in a message flow. It can be a built-in node, a user-defined node, or a subflow node.

A message flow node receives a message, performs a set of actions against the message, and optionally passes the original message, and none or more other messages, to the next node in the message flow.

A message flow node has a fixed number of input and output points known as terminals. You can make connections between the terminals to define the routes that a message can take through a message flow.

This test auto discovers the message flow nodes in the target IBM Integration server and alerts administrators on how well the messages are processed. In addition, this test helps you to review the time taken by each node to process the input messages and the time taken by the CPU of each node to process the input messages. This way, administrators may be alerted to processing bottlenecks on the node that is currently experiencing slowdowns in processing the messages.

Purpose	Auto discovers the message flow nodes in the target IBM Integration server and alerts administrators on how well the messages are processed. In addition, this test helps you to review the time taken by each node to process the input messages and the time taken by the CPU of each node to process the input messages
Target of the test	An IBM Integration Bus
Agent deploying the test	An internal/remote agent

<table>
<tr><td>Configurable parameters for the test</td><td colspan="3">1. TEST PERIOD - How often should the test be executed
2. HOST - The host for which the test is to be configured
3. PORT – The port on which the specified host listens
4. JNDI NAMESPACE LOCATION – Indicate where the JNDI namespace has been created. For this, select one of the following options: File or LDAP. Select File if the JNDI namespace is created in the same location as that of the IIB server.
5. SSL - If you have chosen LDAP as the JNDI NAMESPACE LOCATION wherein the LDAP server communicating with the IIB server that is to be monitored is an SSL-enabled server, then set the SSL flag to Yes. By default, this flag is set to No.
6. USER DN and PASSWORD – Here, provide the credentials of the Active Directory user created as explained in Section 1.1.4.1, if the target IIB server is monitored in an agentless manner. If the IIB server is to be monitored in an agent based manner, specify none against these parameters.
7. CONFIRM PASSWORD – Confirm the PASSWORD by retyping it in this text box.
8. JNDI PROVIDER URL - Specify the URL that was specified as the provider URL while creating the initial context. To know the Provider URL, refer to Figure 7 if you choose to monitor the target IIB server in an agent based manner and Figure 34 if you chose to monitor the target IIB server in an agentless manner.
9. TOPICCONNECTIONFACTORY – Specify the name of the connection factory in this text box. If you have chosen to monitor the target IIB server in an agentless manner, refer to Section 1.1.3.2.1 and if you have chosen to monitor the target IIB server in an agentless manner, refer to Section 1.1.4.3.1.
10. JMS RESOURCE STATS – Specify the name of the JMS destination that you have created for storing the Resource statistics. If you chose to monitor the target IIB server in an agentless manner, refer to Section 1.1.4.3.2 and if you choose to monitor the target IIB server in an agent based manner, refer to Section 1.1.3.2.2.
11. JMS FLOW STATS – Specify the name of the JMS destination that you have created for storing the Message flow statistics. If you chose to monitor the target IIB server in an agentless manner, refer to Section 1.1.4.3.2 and if you choose to monitor the target IIB server in an agent based manner, refer to Section 1.1.3.2.2.</td></tr>
<tr><td>ll Outputs of the test</td><td colspan="3">One set of results for each Execution Group: Message flow: Message flow node of the IBM Integration Bus that is to be monitored</td></tr>
<tr><td rowspan="3">Measurements made by the test</td><td>Measurement</td><td>Measurement Unit</td><td>Interpretation</td></tr>
<tr><td>Average elapsed time:
Indicates the average time taken by this node to process the input messages.</td><td>Secs</td><td>Comparing the value of this measure across the message flow nodes helps you identify the node that is taking too long to process the input messages.</td></tr>
<tr><td>Maximum elapsed time:
Indicates the maximum time taken by this node to process an input message.</td><td>Secs</td><td></td></tr>
</table>

	Average CPU time: Indicates the average time taken by the CPU of this node to process the input messages.	Secs	The value of this measure should be equal to or less than the value of the *Average elapsed time* measure.
	Maximum CPU time: Indicates the maximum time taken by the CPU of this node to process an input message.	Secs	
	Invocation rate: Indicates the rate at which the messages i.e., invocations are processed by this node.	Invocations/sec	Comparing the value of this measure helps you to identify the node that is currently experiencing slowdowns in processing messages.

2.4.2 Message flow statistics Test

A message flow is a sequence of processing steps that run in the broker when an input message is received.

You define a message flow in the IBM® Integration Toolkit by including a number of message flow nodes, each of which represents a set of actions that define a processing step. The way in which you join the message flow nodes together determine which processing steps are carried out, in which order, and under which conditions. The path that you create between one node and another is known as a connection.

A message flow must include an input node that provides the source of the messages that are processed. You can process the message in one or more ways, and optionally deliver it through one or more output nodes. The message is received as a bit stream, and is converted by a parser into a tree structure that is used internally in the message flow. Before the message is delivered to a final destination, it is converted back into a bit stream. When you want to exchange messages between multiple applications, you might find that the applications do not understand or expect messages in the same format. You must provide some processing between the sending and receiving applications to ensure that both can continue to work unchanged, but can exchange messages successfully. When you want to run a message flow to process messages, you deploy it to a broker, where it is run in an integration server. When the target environment, consists of too many message flows that need to be monitored, administrators may often find it difficult to identify the message flow that is experiencing sudden slowdwons in processing the messages. The **Message flow statisitcis** test helps administrators to tackle this problem!

This test auto discovers the message flows in the target IBM Integration Bus and reports the time taken to process the messages. In addition, this test reports the time taken by the CPU to process the messages and the processing rate for the messages. Administrators may also be alerted to the message flows that are frequently error-prone.

Purpose	Auto discovers the message flows in the target IBM Integration Bus and reports the time taken to process the messages. In addition, this test reports the time taken by the CPU to process the messages and the processing rate for the messages. Administrators may also be alerted to the message flows that are frequently error-prone.
Target of the test	An IBM Integration Bus
Agent deploying the test	An internal/remote agent

<table>
<tr><td>Configurable parameters for the test</td><td colspan="3">1. TEST PERIOD - How often should the test be executed
2. HOST - The host for which the test is to be configured
3. PORT – The port on which the specified host listens
4. JNDI NAMESPACE LOCATION – Indicate where the JNDI namespace has been created. For this, select one of the following options: File or LDAP. Select File if the JNDI namespace is created in the same location as that of the IIB server.
5. SSL - If you have chosen LDAP as the JNDI NAMESPACE LOCATION wherein the LDAP server communicating with the IIB server that is to be monitored is an SSL-enabled server, then set the SSL flag to Yes. By default, this flag is set to No.
6. USER DN and PASSWORD – Here, provide the credentials of the Active Directory user created as explained in Section 1.1.4.1, if the target IIB server is monitored in an agentless manner. If the IIB server is to be monitored in an agent based manner, specify none against these parameters.
7. CONFIRM PASSWORD – Confirm the PASSWORD by retyping it in this text box.
8. JNDI PROVIDER URL - Specify the URL that was specified as the provider URL while creating the initial context. To know the Provider URL, refer to Figure 7 if you choose to monitor the target IIB server in an agent based manner and Figure 34 if you chose to monitor the target IIB server in an agentless manner.
9. TOPICCONNECTIONFACTORY – Specify the name of the connection factory in this text box. If you have chosen to monitor the target IIB server in an agentless manner, refer to Section 1.1.3.2.1 and if you have chosen to monitor the target IIB server in an agentless manner, refer to Section 1.1.4.3.1.
10. JMS RESOURCE STATS – Specify the name of the JMS destination that you have created for storing the Resource statistics. If you chose to monitor the target IIB server in an agentless manner, refer to Section 1.1.4.3.2 and if you choose to monitor the target IIB server in an agent based manner, refer to Section 1.1.3.2.2.
11. JMS FLOW STATS – Specify the name of the JMS destination that you have created for storing the Message flow statistics. If you chose to monitor the target IIB server in an agentless manner, refer to Section 1.1.4.3.2 and if you choose to monitor the target IIB server in an agent based manner, refer to Section 1.1.3.2.2.</td></tr>
<tr><td>ll Outputs of the test</td><td colspan="3">One set of results for each Execution Group: Message flow on the IBM Integration Bus that is to be monitored</td></tr>
<tr><td rowspan="3">Measurements made by the test</td><td>Measurement</td><td>Measurement Unit</td><td>Interpretation</td></tr>
<tr><td>Average elapsed time:
Indicates the average time taken by this message flow to process input messages.</td><td>Secs</td><td>Comparing the value of this message across the message flows helps you identify the message flow that is taking too long to process the input messages.</td></tr>
<tr><td>Maximum elapsed time:
Indicates the maximum time taken by this message flow to process a message.</td><td>Secs</td><td></td></tr>
</table>

	Average CPU time: Indicates the average time taken by the CPU of this message flow to process the input messages.	Secs	Comparing the value of this message across the message flows helps you identify the message flow that is taking too long to process the input messages.
	Maximum CPU time: Indicates the maximum time taken by the CPU of this message flow to process an input message.	Secs	
	Input message rate: Indicates the rate at which input messages are processed in this message flow.	Messages/sec	
	Message data rate: Indicates the number of bytes of the incoming messages processed by all the nodes of this message flow per second.	Bytes/sec	
	Number of maximum thread events: Indicates the number of times all the threads of this message flow was utilized.	Number	
	MQ errors: Indicates the number of errors such as MQGET errors or HTTPInput errors that occurred in this message flow.	Number	Ideally, the value of this measure is zero. This measure includes errors such as the conversion errors that occurred when the message is received from the queue. Comparing the value of this measure across message flows will help administrators to identify the message flow that was error prone.
	Error message rate: Indicates the rate at which the error messages were received in this message flow.	Messages/sec	Ideally, the value of this measure should be zero. These errors include exceptions that are thrown downstream of the input node, and errors that are detected by the input node after it successfully retrieves the message from the queue, but before it propagates it to the output terminal (for example, a format error).

	Processing error rate: Indicates the rate at which errors occurred while the nodes were processing the messages in this message flow.	Messages/sec	Ideally, the value of this measure should be zero.
	Commit rate: Indicates the rate at which commits happened on the transactions of this message flow.	Commits/sec	
	Backout rate: Indicates the rate at which transactions backout occurred in this message flow.	Backouts/sec	

2.4.3 Thread statistics Test

In the message flow execution environment, the message flow is thread-safe. You can run message flows concurrently on many operating system threads, without having to consider serialization issues.

Each input message that passes through a message flow for processing by a series of nodes executes on a single thread; it is processed only by the thread that received it. If you want to increase the throughput of a message flow, you can increase the number of threads that are assigned to that message flow. The memory requirements of an integration server are not unduly affected by running message flows on more operating system threads.

This test monitors each thread on the message flow and reports the time taken to process the incoming messages. Using this test, you can figure out how well the messages are processed and eventually identify the thread that is the slowest in processing the messages.

Purpose	Monitors each thread on the message flow and reports the tim taken to process the incoming messages. Using this test, you can figure out how well the messages are processed and eventually identify the thread that is the slowest in processing the messages
Target of the test	An IBM Integration Bus
Agent deploying the test	An internal/remote agent

<table>
<tr><td>Configurable parameters for the test</td><td colspan="3">1. TEST PERIOD - How often should the test be executed
2. HOST - The host for which the test is to be configured
3. PORT – The port on which the specified host listens
4. JNDI NAMESPACE LOCATION – Indicate where the JNDI namespace has been created. For this, select one of the following options: File or LDAP. Select File if the JNDI namespace is created in the same location as that of the IIB server.
5. SSL - If you have chosen LDAP as the JNDI NAMESPACE LOCATION wherein the LDAP server communicating with the IIB server that is to be monitored is an SSL-enabled server, then set the SSL flag to Yes. By default, this flag is set to No.
6. USER DN and PASSWORD – Here, provide the credentials of the Active Directory user created as explained in Section 1.1.4.1, if the target IIB server is monitored in an agentless manner. If the IIB server is to be monitored in an agent based manner, specify none against these parameters.
7. CONFIRM PASSWORD – Confirm the PASSWORD by retyping it in this text box.
8. JNDI PROVIDER URL - Specify the URL that was specified as the provider URL while creating the initial context. To know the Provider URL, refer to Figure 7 if you choose to monitor the target IIB server in an agent based manner and Figure 34 if you chose to monitor the target IIB server in an agentless manner.
9. TOPICCONNECTIONFACTORY – Specify the name of the connection factory in this text box. If you have chosen to monitor the target IIB server in an agentless manner, refer to Section 1.1.3.2.1 and if you have chosen to monitor the target IIB server in an agentless manner, refer to Section 1.1.4.3.1.
10. JMS RESOURCE STATS – Specify the name of the JMS destination that you have created for storing the Resource statistics. If you chose to monitor the target IIB server in an agentless manner, refer to Section 1.1.4.3.2 and if you choose to monitor the target IIB server in an agent based manner, refer to Section 1.1.3.2.2.
11. JMS FLOW STATS – Specify the name of the JMS destination that you have created for storing the Message flow statistics. If you chose to monitor the target IIB server in an agentless manner, refer to Section 1.1.4.3.2 and if you choose to monitor the target IIB server in an agent based manner, refer to Section 1.1.3.2.2.</td></tr>
<tr><td>ll Outputs of the test</td><td colspan="3">One set of results for each Execution Group:Message flow:Thread of the IBM Integration Bus that is to be monitored</td></tr>
<tr><td>Measurements made by the</td><td>Measurement</td><td>Measurement Unit</td><td>Interpretation</td></tr>
</table>

test	**Average elapsed time:** Indicates the average time elapsed to process input messages by this thread per second.	Messages/sec	A high value for this measure is a cause of concern as this may be due to the processing of messages that are currently in an infinite loop. Administrators may need to check for such messages so as to maintain the value of this measure within optimal limits. Comparing the value of this measure across the threads helps you in identifying the thread that is taking too long to process the incoming messages.
	Average CPU time: Indicates the average time taken by the CPU to process the input messages.	Secs	
	Input message rate: Indicates the time taken by this thread to process the input messages.	Secs	Comparing the value of this measure across the threads helps you in identifying the thread that is taking too long to process the incoming messages.
	Message data rate: Indicates the rate at which bytes of the input messages are processed by this thread.	Bytes/sec	Comparing the value of this measure across the threads will help you identify the thread that is experiencing slowdowns in processing the messages.

3 Conclusion

This document has described in detail the monitoring paradigm used and the measurement capabilities of the eG Enterprise suite of products with respect to the **IBM Integration Bus**. For details of how to administer and use the eG Enterprise suite of products, refer to the user manuals.

We will be adding new measurement capabilities into the future versions of the eG Enterprise suite. If you can identify new capabilities that you would like us to incorporate in the eG Enterprise suite of products, please contact support@eginnovations.com. We look forward to your support and cooperation. Any feedback regarding this manual or any other aspects of the eG Enterprise suite can be forwarded to feedback@eginnovations.com.