Monitoring JBoss Application Server

eG Innovations Product Documentation

www.eginnovations.com

Total Performance Visibility

Table of Contents

Table of Figures

Chapter 1: Introduction

The JBoss Application Server is a widely used Java application server that provides a J2EE certified platform for developing and deploying enterprise Java applications, web applications, and portals, and also offers extended enterprise services such as clustering, caching, and persistence.

To monitor the JBoss application server, eG requires administrators to deploy two specialized components on the target server. These components are, the **eG MBean Service** component, and the **eG Web Component**.

- **eG MBean Service Component:** This service is started by JBoss as soon as this component is deployed to the server. The main purpose of this component is to get the MBean server instance running on the JBoss server. It also provides static methods which use the MBean server instance to gather relevant statistics from the JBoss server.
- **eG Web Component:** This web component consists of one servlet. The eG agent will connect to this servlet by passing relevant arguments. The servlet will then invoke the corresponding static method of the MBean service component and return the result to the eG agent. The agent will then compute the required metrics and upload the same the eG manager.

In order to enable the eG agents to extract statistics from the JBoss server, the above-mentioned components need to be deployed on the JBoss server. The steps for deploying these components have been discussed in **How to Monitor JBoss Application Server Using eG Enterprise?** chapter.

Chapter 2: How to Monitor JBoss Application Server Using eG Enterprise?

A JBoss application server (version 4.0 and 4.2.3) can be monitored in an agent-based or an agentless manner using eG Enterprise. Before attempting JBoss monitoring, the JBoss application server should be configured to work with the eG agent. The following sections describe how to configure a JBoss application server to work with eG agent.

2.1 Configuring the JBoss server to work with the eG Agent

To monitor the JBoss server, eG requires administrators to deploy two specialized components on the target server. These components are, the **eG MBean Service** component, and the **eG Web** Component. In order to enable the eG agents to extract statistics from the JBoss server, the above-mentioned components need to be deployed on the JBoss server. To achieve this, do the following:

1. To deploy the **eG MBean Service** component on a JBoss server (prior to JBoss v4.2), a file named **EgJBossAgent.sar** will have to be copied from the <EG_INSTALL_DIR>\lib directory of the eG agent (running Windows; on Unix, this will be the **/opt/egurkha/lib** directory) to the **<JBOSS_ INSTALL_ DIR>\server\<JBOSS_ APPLICATION_ SERVER_ INSTANCE>\deploy** directory on the JBoss server to be monitored. For example, if the application server instance of the target JBoss server is 'default', and the server is installed in the 'D:\JBoss' directory, then the file will have to be copied to the **D:\JBoss\server\default\deploy** directory.
2. Likewise, to deploy the **eG MBean Service** component on a JBoss server of v4.2 (and above), a file named **EgJBossAgent42.sar** will first have to be copied from the <EG_INSTALL_DIR>\lib directory of the eG agent host (running Windows; on Unix, this will be the **/opt/egurkha/lib** directory) to any temporary folder on the same host. Then, rename the file to **EgJbossAgent.sar**, and then copy it to the **<JBOSS_INSTALL_ DIR>\server\<JBOSS_ APPLICATION_ SERVER_ INSTANCE>\deploy** directory on the JBoss server to be monitored.
3. For the **eG Web** component to be deployed on the JBoss server, you will have to copy a file named **egjboss.war** to the server. This file will be available in the <EG_INSTALL_DIR>\lib directory in Windows installations of the eG agent. On Unix, these files will be available in the **/opt/egurkha/lib** directory.
4. Next, proceed to restart the server. Prior to this, ensure that the server can be accessed using its IP address. This is because, by default, Jboss is bound to the local host. This implies that, by

default, you will not be able to pull out metrics from the JBoss server or access the resources of the server using its IP address. This default setting could cause the eG tests to fail. In order to avoid this, you will have to ensure that the JBoss server is accessible via the IP address. For that, to start/restart the server, run the following command at the command prompt of the server:

run.bat -b ip

Note:

To be effectively monitored by the eG agent, the JBoss server should be of version 4.0 or 4.2.3, and should be executing on JDK 1.5 or higher.

2.2 Configuring the eG Agent to Collect JVM-related Metrics from the JBoss server Ver. 4.0

The *jvm* layer of the *JBoss* monitoring model is associated with tests that report critical statistics related to the *JBoss* server's JVM. These statistics typically reveal the following:

- The count of classes loaded/unloaded (**Java Classes** test)
- JVM thread usage (**JVM Threads** test)
- CPU and memory usage of the JVM (**JVM Cpu Usage** test and **JVM Memory Usage** test)
- The effectiveness of the JVM's garbage collection activity (**JVM Garbage Collections** test)
- The uptime of the JVM (**JVM Uptime** test)
- Whether JMX is currently enabled/disabled on the target WebLogic server (**JMX Connection to JVM** test)
- The count and status of file descriptors (**JVM File Descriptors** test)

These tests connect to the JRE used by the JBoss application server to pull out the above-mentioned metrics. To enable the tests to collect metrics from the JRE, the eG agent executing these tests should be configured to use one of the following methodologies:

- JMX (Java Management Extensions)
- SNMP (Simple Network Management Protocol)

Since both JMX and SNMP support are available for JRE 1.5 and above only, these tests **will work only if the JBoss server being monitored runs JRE 1.5 and above**.

If the target JBoss server indeed uses JRE 1.5 (or above), then, you can proceed to use either JMX or SNMP for connecting to the JRE of the target and pulling out the desired metrics.

If you choose to use JMX for metrics, then the following broad steps need to be followed:

1. First, determine whether the JMX requires no authentication at all, or requires authentication (but no security)
2. If JMX does not require authentication, follow the steps below:
    - Login to the target JBoss server.
    - Edit the *management.properties* file that is used by the JRE of the target JBoss server, and configure the following in it:
    - The JMX remote port
    - Whether JMX is SSL-enabled or not
    - Whether JMX requires authentication or not
    - To know how to configure these, refer to the *Monitoring Java Applications* document.
    - Save the file.

    Note:
    To know how to enable SNMP support for the JRE, refer to the Monitoring Java Applications document.
3. If the JMX requires authentication (but no security), follow the steps below:
    - Login to the target JBoss server. If the server is executing on a Windows host, then, login to the host as a local/domain administrator.
    - Next, copy the *jmxremote.password.template* file in the **<JAVA_HOME>\jre\lib\management** folder to any other location on the host, rename it as as *jmxremote.password*, and then, copy it back to the <JAVA_HOME>\jre\lib\management folder.
    - Next, edit the *jmxremote.password* file and the *jmxremote.access* file to create a user with *read-write* access to the JMX. To know how to create such a user, refer to *Monitoring Java Applications* document.
    - Then, proceed to make the *jmxremote.password* file secure by granting a single user "full access" to that file. To know how to achieve this, refer to the *Monitoring Java Applications* document.
    - Edit the *management.properties* file that is used by the JRE of the target JBoss server, and configure the following in it:
        - The JMX remote port
        - Whether JMX is SSL-enabled or not

- Whether JMX requires authentication or not
- The full path to the *jmxremote.access* file
- The full path to the *jmxremote.password* file
- To know how to configure these, refer to the *Monitoring Java Applications* document.
- Then, save the file.
- Then, edit the **<JBOSS_HOME>\bin\run.bat**file.
- Add the following line in it:

```
set JAVA_OPTS=-Dcom.sun.management.config.file=<<Path to management.properties
file used by the JRE of the target JBoss server>
```

- Finally, restart the server.

Then, proceed to enable the JVM-related tests, and configure them using the instructions provided in the *Monitoring Java Application Servers* document. To enable one/more tests, go to the **ENABLE / DISABLE TESTS** page using the menu sequence : Agents -> Tests -> Enable/Disable, pick *JBoss* as the **Component type,** *Performance* as the **Test type**, choose the tests from the **DISABLED TESTS** list, and click on the **>>** button to move the tests to the **ENABLED TESTS** list. Finally, click the **Update** button.

Once you have configured the JBoss application server to work with the eG agent, manage the JBoss application server component using eG administrative interface. The procedure for achieving this has been explained in Section **2.3**.

2.3 Managing the JBoss Application server

eG Enterprise can automatically discover the JBoss server in the environment and also lets you to manually add the JBoss Server for monitoring if the server is not auto-discovered. The following steps explain you how to manually add the JBoss server for monitoring.

1. Log into the eG administrative interface.
2. If a JBoss Server is already been discovered, then directly proceed towards managing it using the **COMPONENTS – MANAGE/UNMANAGE** page.
3. However, if it is yet to be discovered, then run discovery (Infrastructure -> Components -> Discover) to get it discovered or add the component manually using the **COMPONENTS** page (Infrastructure -> Components -> Add/Modify). Remember that components manually added are

managed automatically. Discovered components, however, are managed using the **COMPONENTS – MANAGE / UNMANAGE** page.

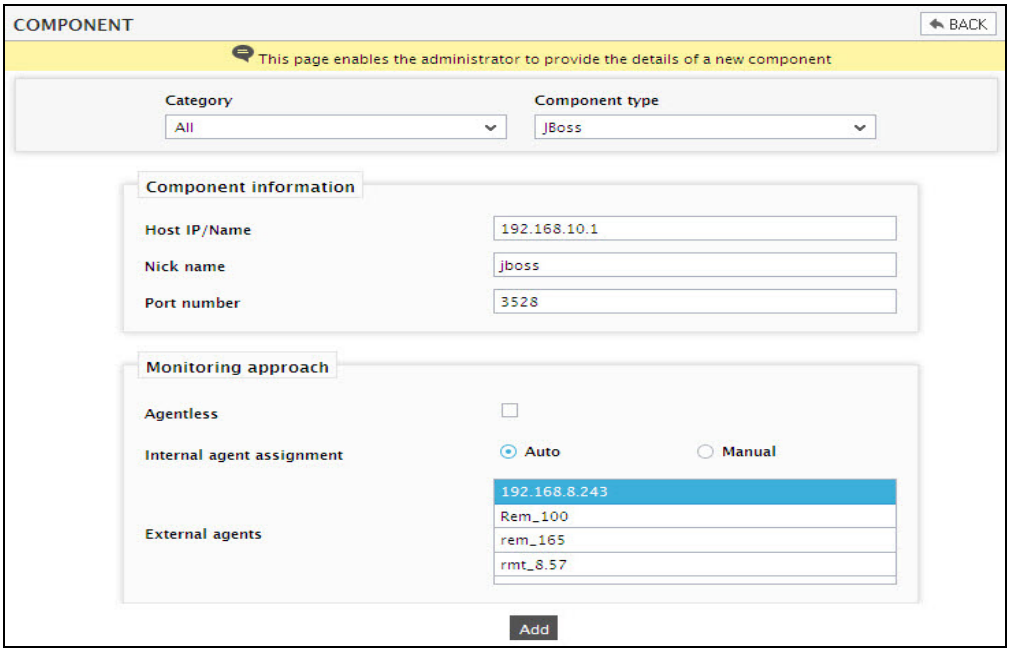

Figure 2.1: Adding a JBoss server

4. Next, try to sign out of the eG administrative interface. Upon doing so, a list of unconfigured tests will appear prompting you to configure the tests pertaining to JBoss server.

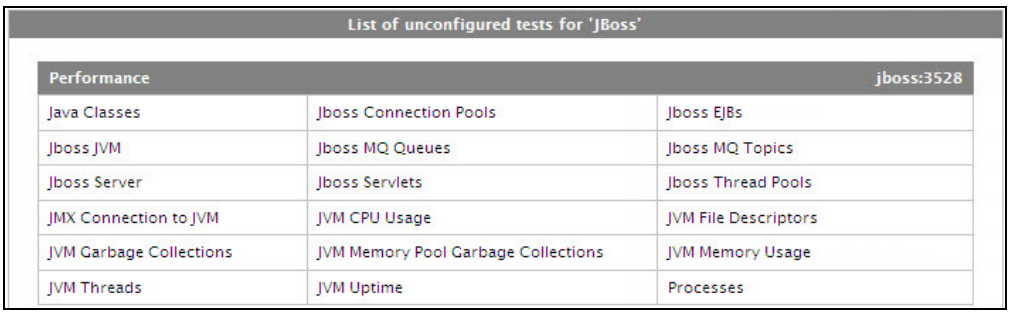

Figure 2.2: List of unconfigured tests for JBoss

5. Then, signout of the eG administrative interface. The list of unconfigured tests page will then reappear. This time, pick the **Java Classes** test for configuration. This test reports the number of classes loaded/unloaded from the memory.
6. Finally, click the **Update** button, and then signout of the eG administrative interface.

Chapter 3: Monitoring JBoss Application Servers

Once the components are deployed on the JBoss server, the eG agent periodically executes tests on the server to capture the statistics collected by the components, and reports them to the eG manager. These tests are mapped to specific layers of the JBoss application server's layer model (see Figure 3.1).

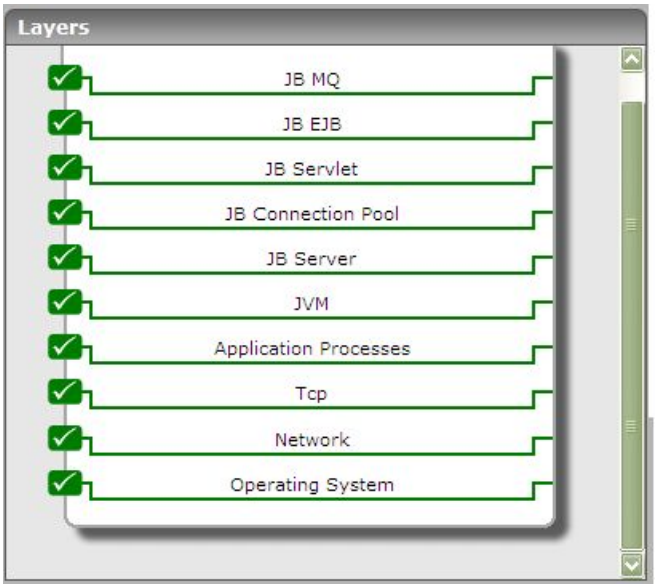

Figure 3.1: The layer model of a JBoss application server

Note:

The JBoss monitoring model provided by eG Enterprise (see Figure 3.1) can be used to monitor a JBoss server (versions 4.0 and 4.2.3), in an agent-based or an agentless manner.

The measures reported by the tests enable JBoss administrators to make accurate performance judgments with respect to the following:

JVM heap monitoring	• How much memory has been allocated to the JVM heap? • Is adequate free memory available in the JVM heap?
Request processing capability Monitoring	• Is the server taking too long to process requests? • Are too many errors encountered by the server during request processing?

	• Is the server overloaded with requests?
Thread pool monitoring	• How many threads are currently busy? Does the server appear to be handling too much load? • Are there adequate spare threads in the pool for future processing requirements? • Does the thread pool require resizing?
Connection pool monitoring	• Are enough connections available in the connection pool, or do more connections have to be allotted to the pool?
Servlet monitoring	• How frequently was the servlet invoked? • Does the servlet take too long to execute?
EJB monitoring	• How frequently are EJBs created/removed? • Are there any EJBs that are ready to service clients? • How many EJBs are currently in the EJB pool?
Messaging service monitoring	• Are too many messages have been enqueued? • How quickly does the queue process messages within?
Topic monitoring	• Does the topic take too long to process messages? • What type of messages are currently available in the topic - durable or non-durable? • How many durable subscribers are there to a topic?

The sections to come will indulge in an elaborate discussion of the top 6 layers of the layer model depicted by Figure 3.1. As the other layers have been dealt with in great detail in the *Monitoring Unix and Windows Servers* document.

3.1 The JVM Layer

This layer collectively reports the resource usage and overall health of the JBoss JVM.

Figure 3.2: The tests mapped to the JVM layer

All the tests displayed in Figure 3.2 have been dealt with in the *Monitoring Java Applications* document.

3.2 The JB Server Layer

The tests associated with the **JB Server** layer (see Figure 3.3) measure the request processing capability of the JBoss server by monitoring critical parameters such as thread pool usage and memory heap utilization.

Figure 3.3: The tests associated with the JB Server layer

3.2.1 Jboss JVM Test

This test monitors the heap usage and thread usage of the Java Virtual Machine (JVM) on which the JBoss application server runs.

Target of the test : A JBoss application server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for every JBoss server monitored.

Configurable parameters for the test

Parameter	Description
Test period	How often should the test be executed.
Host	The host for which the test is to be configured.
Port	The port number at which the specified host listens to.
Measurement Mode	This test can extract metrics from JBoss using either of the following mechanisms: • By deploying the **EgJBossAgent.sar** and **egjboss.war** files on the target JBoss server; • By contacting the Java runtime (JRE) of JBoss viaJMX To configure the test to usethe **EgJBossAgent.sar** and **egjboss.war** files, first, select the **War File** option. Then, refer to the **How to Monitor JBoss Application Server Using eG Enterprise?** chapter to know how to deploy these files on the target JBoss server. On the other hand, if you want the test to use JMX instead, then first, select the **jmx** option. Then, follow the procedure detailed in the **How to Monitor JBoss Application Server Using eG Enterprise?** chapter to configure the test to use **jmx**. By default, the **jmx** option is chosen here.
JNDIName	This parameter appears only if the Measurement Mode is set to **jmx**. The JNDIName is a lookup name for connecting to the JMX connector. By default, this is *jmxrmi*. If you have registered the JMX connector in the RMI registry using a different lookup name, then you can change this default value to reflect the same.
JMX Remote Port	This parameter appears only if the Measurement Mode is set to **jmx**. Here, specify the port at which the **jmx** listens for requests from remote hosts. Ensure that you specify the same port that you configured in the *management.properties* file of the target JBoss server (refer to the **How to Monitor JBoss Application Server Using eG Enterprise?** chapter for more details).
JMX User, JMX Password, and Confirm Password	These parameters appear only if the Measurement Mode is set to **jmx**. If JMX requires **authentication only** (but no security), then ensure that the JMX User and JMX Password parameters are configured with the credentials of a user with *read-write* access to JMX. To know how to create this user, refer to the **How to Monitor JBoss Application Server Using eG Enterprise?** chapter. Confirm the password by retyping it in the Confirm Password text box.

Parameter	Description
SSL	This parameter appears only if the Measurement Mode is set to **War File**. Indicate **Yes** if the JBoss server is SSL-enabled.
URL	This parameter appears only if the Measurement Mode is set to **War File**. Specify the URL to be accessed to collect metrics pertaining to the JBoss server. By default, this test connects to a managed JBoss server and attempts to obtain the metrics of interest by accessing the local Mbeans of the server. Accordingly, by default, this parameter is set to *http://<JBossServerIP>:<JbossServerPort>*.
JMX Provider	This parameter appears only if the Measurement Mode is set to **jmx**. This test uses a JMX Provider to access the MBean attributes of the JBoss server and collect metrics. Specify the package name of this JMX Provider here. By default, this is set to *com.sun.jmx.remote.protocol*.
Timeout	Specify the duration (in seconds) for which this test should wait for a response from the JBoss server. If there is no response from the server beyond the configured duration, the test will timeout. By default, this is set to *240* seconds.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Free memory	Indicates the memory available in the JVM heap of the JBoss application server.	MB	If this value decreases consistently, then try increasing the memory heap of the JVM.
Used memory	Indicates the amount of JVM heap memory used.	MB	
Total memory	Indicates the total memory allocated to the JVM.	MB	
Active threads	Indicates the total number of threads that are currently active in the JBoss server.	Number	

3.2.2 Jboss Server Test

This test monitors the requests handled by the JBoss server.

Target of the test : A JBoss application server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for every JBoss server monitored.

Configurable parameters for the test

Parameter	Description
Test period	How often should the test be executed.
Host	The host for which the test is to be configured.
Port	The port number at which the specified host listens to.
Measurement Mode	This test can extract metrics from JBoss using either of the following mechanisms: • By deploying the **EgJBossAgent.sar** and **egjboss.war** files on the target JBoss server; • By contacting the Java runtime (JRE) of JBoss viaJMX To configure the test to usethe **EgJBossAgent.sar** and **egjboss.war** files, first, select the **War File** option. Then, refer to the **How to Monitor JBoss Application Server Using eG Enterprise?** chapter to know how to deploy these files on the target JBoss server. On the other hand, if you want the test to use JMX instead, then first, select the **jmx** option. Then, follow the procedure detailed in the **How to Monitor JBoss Application Server Using eG Enterprise?** chapter to configure the test to use **jmx**. By default, the **jmx** option is chosen here.
JNDIName	This parameter appears only if the Measurement Mode is set to **jmx**. The JNDIName is a lookup name for connecting to the JMX connector. By default, this is *jmxrmi*. If you have registered the JMX connector in the RMI registry using a different lookup name, then you can change this default value to reflect the same.
JMX Remote Port	This parameter appears only if the Measurement Mode is set to **jmx**. Here, specify the port at which the **jmx** listens for requests from remote hosts. Ensure that you specify the same port that you configured in the *management.properties* file of the target JBoss server (refer to the **How to Monitor JBoss Application Server Using eG Enterprise?** chapter for more details).
JMX User, JMX Password, and Confirm Password	These parameters appear only if the Measurement Mode is set to **jmx**. If JMX requires **authentication only** (but no security), then ensure that the JMX User and JMX Password parameters are configured with the credentials of a user with *read-write* access to JMX. To know how to create this user, refer to the **How to Monitor JBoss**

Parameter	Description
	Application Server Using eG Enterprise? chapter. Confirm the password by retyping it in the Confirm Password text box.
SSL	This parameter appears only if the Measurement Mode is set to **War File**. Indicate **Yes** if the JBoss server is SSL-enabled.
URL	This parameter appears only if the Measurement Mode is set to **War File**. Specify the URL to be accessed to collect metrics pertaining to the JBoss server. By default, this test connects to a managed JBoss server and attempts to obtain the metrics of interest by accessing the local Mbeans of the server. Accordingly, by default, this parameter is set to *http://<JBossServerIP>:<JbossServerPort>*.
JMX Provider	This parameter appears only if the Measurement Mode is set to **jmx**. This test uses a JMX Provider to access the MBean attributes of the JBoss server and collect metrics. Specify the package name of this JMX Provider here. By default, this is set to *com.sun.jmx.remote.protocol*.
Timeout	Specify the duration (in seconds) for which this test should wait for a response from the JBoss server. If there is no response from the server beyond the configured duration, the test will timeout. By default, this is set to *240* seconds.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Request rate	Indicates the rate of requests to the server during the last measurement period.	Reqs/Sec	With the advent of HTTP/1.1 multiple requests can be transmitted over the same TCP connection. The ratio of requests per connection can provide an idea of the effectiveness of the HTTP 1.1 protocol.
Data transmit rate	Indicates the rate at which the data was transmitted by the server during the last measurement period.	KB/Sec	A large increase in the data transmission rate can be indicative of an increase in the popularity of one or more web sites hosted on the server.
Data received rate	Indicates the rate at which data was received by the server during the last measurement period.	KB/Sec	An increase in this value is indicative of an increase in user requests to the server.
Avg request	Indicates the average time	MSecs	A high value may be an indication of

Measurement	Description	Measurement Unit	Interpretation
processing time	taken by the server to process requests.		slow performance of the server.
Max request processing time	Indicates the maximum time taken by the server to process requests.	MSecs	
Error rate	Indicates the rate at which errors were encountered by the server in the last measurement period.	Errors/Sec	

3.2.3 Jboss Thread Pools Test

This test monitors the thread pools in the JBoss server.

Target of the test : A JBoss application server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for each thread pool configured in the JBoss server.

Configurable parameters for the test

Parameter	Description
Test period	How often should the test be executed.
Host	The host for which the test is to be configured.
Port	The port number at which the specified host listens to.
Measurement Mode	This test can extract metrics from JBoss using either of the following mechanisms: • By deploying the **EgJBossAgent.sar** and **egjboss.war** files on the target JBoss server; • By contacting the Java runtime (JRE) of JBoss viaJMX To configure the test to usethe **EgJBossAgent.sar** and **egjboss.war** files, first, select the **War File** option. Then, refer to the **How to Monitor JBoss Application Server Using eG Enterprise?** chapter to know how to deploy these files on the target JBoss server.

Parameter	Description
	On the other hand, if you want the test to use JMX instead, then first, select the **jmx** option. Then, follow the procedure detailed in the **How to Monitor JBoss Application Server Using eG Enterprise?** chapter to configure the test to use **jmx**. By default, the **jmx** option is chosen here.
JNDIName	This parameter appears only if the Measurement Mode is set to **jmx**. The JNDIName is a lookup name for connecting to the JMX connector. By default, this is *jmxrmi*. If you have registered the JMX connector in the RMI registry using a different lookup name, then you can change this default value to reflect the same.
JMX Remote Port	This parameter appears only if the Measurement Mode is set to **jmx**. Here, specify the port at which the **jmx** listens for requests from remote hosts. Ensure that you specify the same port that you configured in the *management.properties* file of the target JBoss server (refer to the **How to Monitor JBoss Application Server Using eG Enterprise?** chapter for more details).
JMX User, JMX Password, and Confirm Password	These parameters appear only if the Measurement Mode is set to **jmx**. If JMX requires **authentication only** (but no security), then ensure that the JMX User and JMX Password parameters are configured with the credentials of a user with *read-write* access to JMX. To know how to create this user, refer to the **How to Monitor JBoss Application Server Using eG Enterprise?** chapter. Confirm the password by retyping it in the Confirm Password text box.
SSL	This parameter appears only if the Measurement Mode is set to **War File**. Indicate **Yes** if the JBoss server is SSL-enabled.
URL	This parameter appears only if the Measurement Mode is set to **War File**. Specify the URL to be accessed to collect metrics pertaining to the JBoss server. By default, this test connects to a managed JBoss server and attempts to obtain the metrics of interest by accessing the local Mbeans of the server. Accordingly, by default, this parameter is set to *http://<JBossServerIP>:<JbossServerPort>*.
JMX Provider	This parameter appears only if the Measurement Mode is set to **jmx**. This test uses a JMX Provider to access the MBean attributes of the JBoss server and collect metrics. Specify the package name of this JMX Provider here. By default, this is set to *com.sun.jmx.remote.protocol*.
Timeout	Specify the duration (in seconds) for which this test should wait for a response from the JBoss server. If there is no response from the server beyond the configured duration, the test will timeout. By default, this is set to *240* seconds.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Current busy threads	Indicates the number of threads that are currently busy.	Number	The value of this measure serves as a good indicator of server workload.
Current thread count	Indicates the number of threads currently spawned in the JVM.	Number	
Min spare threads	Indicates the minimum number of spare threads.	Number	
Max spare threads	Indicates the maximum number of spare threads.	Number	
Max threads	Indicates the maximum number of threads that this pool can contain.	Number	If the number of threads in a pool grows close to the value of this measure, it indicates that the number of threads in the pool needs to be increased for effective operation of this application. Alternatively, you may also consider changing the maximum number of threads that a pool can contain. However, exercise caution when altering the maximum thread count, as a very high thread count can cause the app to slowdown from excessive memory usage. Likewise, if the maximum thread count is set too low, it will cause requests to block or timeout.

3.3 The JB Connection Pool Layer

This layer of the JBoss application server monitors the connection pools configured in the JBoss application server, and indicates whether the pools have been adequately sized and are being effectively utilized.

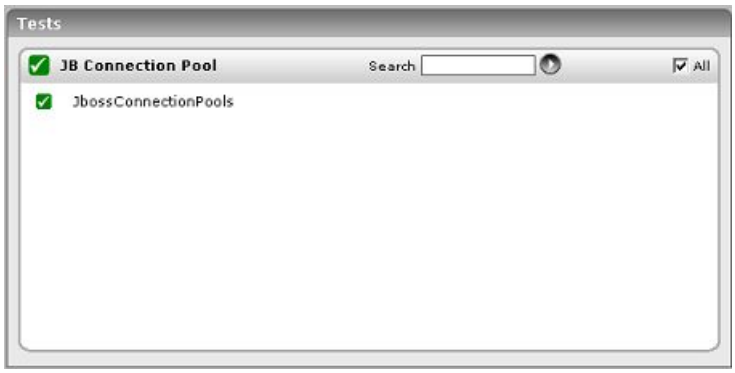

Figure 3.4: The test associated with the JB Connection Pool layer

3.3.1 Jboss Connection Pools Test

This test monitors the various connection pools configured in the JBoss server, and reveals how well the pools have been used.

Target of the test : A JBoss application server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for every JBoss server monitored.

Configurable parameters for the test

Parameter	Description
Test period	How often should the test be executed.
Host	The host for which the test is to be configured.
Port	The port number at which the specified host listens to.
Measurement Mode	This test can extract metrics from JBoss using either of the following mechanisms: • By deploying the **EgJBossAgent.sar** and **egjboss.war** files on the target JBoss server; • By contacting the Java runtime (JRE) of JBoss viaJMX To configure the test to usethe **EgJBossAgent.sar** and **egjboss.war** files, first, select the **War File** option. Then, refer to the **How to Monitor JBoss Application Server Using eG Enterprise?** chapter to know how to deploy these files on the target JBoss server.

Parameter	Description
	On the other hand, if you want the test to use JMX instead, then first, select the **jmx** option. Then, follow the procedure detailed in the **How to Monitor JBoss Application Server Using eG Enterprise?** chapter to configure the test to use **jmx**. By default, the **jmx** option is chosen here.
JNDIName	This parameter appears only if the Measurement Mode is set to **jmx**. The JNDIName is a lookup name for connecting to the JMX connector. By default, this is *jmxrmi*. If you have registered the JMX connector in the RMI registry using a different lookup name, then you can change this default value to reflect the same.
JMX Remote Port	This parameter appears only if the Measurement Mode is set to **jmx**. Here, specify the port at which the **jmx** listens for requests from remote hosts. Ensure that you specify the same port that you configured in the *management.properties* file of the target JBoss server (refer to the **How to Monitor JBoss Application Server Using eG Enterprise?** chapter for more details).
JMX User, JMX Password, and Confirm Password	These parameters appear only if the Measurement Mode is set to **jmx**. If JMX requires **authentication only** (but no security), then ensure that the JMX User and JMX Password parameters are configured with the credentials of a user with *read-write* access to JMX. To know how to create this user, refer to the **How to Monitor JBoss Application Server Using eG Enterprise?** chapter. Confirm the password by retyping it in the Confirm Password text box.
SSL	This parameter appears only if the Measurement Mode is set to **War File**. Indicate **Yes** if the JBoss server is SSL-enabled.
URL	This parameter appears only if the Measurement Mode is set to **War File**. Specify the URL to be accessed to collect metrics pertaining to the JBoss server. By default, this test connects to a managed JBoss server and attempts to obtain the metrics of interest by accessing the local Mbeans of the server. Accordingly, by default, this parameter is set to *http://<JBossServerIP>:<JbossServerPort>*.
JMX Provider	This parameter appears only if the Measurement Mode is set to **jmx**. This test uses a JMX Provider to access the MBean attributes of the JBoss server and collect metrics. Specify the package name of this JMX Provider here. By default, this is set to *com.sun.jmx.remote.protocol*.
Timeout	Specify the duration (in seconds) for which this test should wait for a response from the JBoss server. If there is no response from the server beyond the configured duration, the test will timeout. By default, this is set to *240* seconds.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Connections created	Indicates the total number of connections created since the start of the connection pool.	Number	
Current connections	Indicates the number of connections currently in the pool.	Number	
Connections in use	Indicates the number of connections that are currently in use.	Number	If the value of this measure reaches that of the *Current connections* measure, then it indicates that the pool requires resizing.
Connections destroyed	Indicates the number of connections that were destroyed since the start of the server.	Number	
Max connections in use	Indicates the high water mark of the connections in use count.	Number	
Available connections	Indicates the number of connections that are currently available in the pool for clients to use.	Number	Ideally, this value should be high.
Min connection size	Indicates the minimum value of the number of connections in the pool since the start of the pool.	Number	
Max connection size	Indicates the maximum value of the number of connections in the pool since the start of the pool.	Number	

3.4 The JB Servlet Layer

The **JB Servlet** layer executes the **JbossServlets** test (see Figure 3.5) that monitors the performance of the servlets deployed on the JBoss application server.

Figure 3.5: The test associated with the JB Servlet layer

3.4.1 Jboss Servlets Test

This test monitors the servlets deployed on the JBoss server. If the JBoss server hosts a large number of servlets, then managing the individual servlets would be quiet a challenge. Therefore, to enable administrators to manage servlets easily, the eG Enterprise system allows the configuration of servlet groups. The eG agent executing this test will then, by default, report measures for every group that is configured. To configure servlet groups, click on the **Click here** hyperlink in the test configuration page of the JbServletTest.

Target of the test : A JBoss application server

Agent deploying the test : An internal agent

Outputs of the test : By default, the test reports one set of results for each servlet group that is configured using the eG administrative interface.

Configurable parameters for the test

Parameter	Description
Test period	How often should the test be executed.
Host	The host for which the test is to be configured.
Port	The port number at which the specified host listens to.
Measurement Mode	This test can extract metrics from JBoss using either of the following mechanisms:

Parameter	Description
	• By deploying the **EgJBossAgent.sar** and **egjboss.war** files on the target JBoss server; • By contacting the Java runtime (JRE) of JBoss viaJMX To configure the test to usethe **EgJBossAgent.sar** and **egjboss.war** files, first, select the **War File** option. Then, refer to the **How to Monitor JBoss Application Server Using eG Enterprise?** chapter to know how to deploy these files on the target JBoss server. On the other hand, if you want the test to use JMX instead, then first, select the **jmx** option. Then, follow the procedure detailed in the **How to Monitor JBoss Application Server Using eG Enterprise?** chapter to configure the test to use **jmx**. By default, the **jmx** option is chosen here.
JNDIName	This parameter appears only if the Measurement Mode is set to **jmx**. The JNDIName is a lookup name for connecting to the JMX connector. By default, this is *jmxrmi*. If you have registered the JMX connector in the RMI registry using a different lookup name, then you can change this default value to reflect the same.
JMX Remote Port	This parameter appears only if the Measurement Mode is set to **jmx**. Here, specify the port at which the **jmx** listens for requests from remote hosts. Ensure that you specify the same port that you configured in the *management.properties* file of the target JBoss server (refer to the **How to Monitor JBoss Application Server Using eG Enterprise?** chapter for more details).
JMX User, JMX Password, and Confirm Password	These parameters appear only if the Measurement Mode is set to **jmx**. If JMX requires **authentication only** (but no security), then ensure that the JMX User and JMX Password parameters are configured with the credentials of a user with *read-write* access to JMX. To know how to create this user, refer to the **How to Monitor JBoss Application Server Using eG Enterprise?** chapter. Confirm the password by retyping it in the Confirm Password text box.
SSL	This parameter appears only if the Measurement Mode is set to **War File**. Indicate **Yes** if the JBoss server is SSL-enabled.
URL	This parameter appears only if the Measurement Mode is set to **War File**. Specify the URL to be accessed to collect metrics pertaining to the JBoss server. By default, this test connects to a managed JBoss server and attempts to obtain the metrics of interest by accessing the local Mbeans of the server. Accordingly, by default, this parameter is set to *http://<JBossServerIP>:<JbossServerPort>*.
JMX Provider	This parameter appears only if the Measurement Mode is set to **jmx**. This test uses a

Parameter	Description
	JMX Provider to access the MBean attributes of the JBoss server and collect metrics. Specify the package name of this JMX Provider here. By default, this is set to *com.sun.jmx.remote.protocol*.
Timeout	Specify the duration (in seconds) for which this test should wait for a response from the JBoss server. If there is no response from the server beyond the configured duration, the test will timeout. By default, this is set to *240* seconds.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Invocation rate	Indicates the number of times the servlet was invoked per second in the last measurement period.	Invocations/Sec	This is indicative of the load on a servlet.
Avg invocation time	Indicates the average time taken to execute a servlet.	Msecs	
Max invocation time	Indicates the maximum time taken to execute a servlet.	Msecs	

3.5 The JB EJB Layer

The **JB EJB** layer executes the **JbossEjbs** test (see Figure 3.6), which evaluates the performance of the Enterprise Java Beans (EJBs) deployed on the JBoss application server.

Figure 3.6: The test associated with the JB EJB layer

3.5.1 Jboss Ejbs Test

This test monitors the EJBs deployed on the JBoss server. If the JBoss server hosts a large number of EJBs, then managing the individual EJBs would be quiet a challenge. Therefore, to enable administrators to manage EJBs easily, the eG Enterprise system allows the configuration of EJB groups. The eG agent executing this test will then, by default, report measures for every group that is configured. To configure EJB groups, click on the **Click here** hyperlink in the test configuration page of the **Jboss Ejbs** test.

Target of the test : A JBoss application server

Agent deploying the test : An internal agent

Outputs of the test : By default, the test reports one set of results for each EJB group that is configured using the eG administrative interface.

Configurable parameters for the test

Parameter	Description
Test period	How often should the test be executed.
Host	The host for which the test is to be configured.
Port	The port number at which the specified host listens to.
Measurement Mode	This test can extract metrics from JBoss using either of the following mechanisms: • By deploying the **EgJBossAgent.sar** and **egjboss.war** files on the target JBoss server; • By contacting the Java runtime (JRE) of JBoss viaJMX To configure the test to usethe **EgJBossAgent.sar** and **egjboss.war** files, first, select the **War File** option. Then, refer to the **How to Monitor JBoss Application Server Using eG Enterprise?** chapter to know how to deploy these files on the target JBoss server. On the other hand, if you want the test to use JMX instead, then first, select the **jmx** option. Then, follow the procedure detailed in the **How to Monitor JBoss Application Server Using eG Enterprise?** chapter to configure the test to use **jmx**. By default, the **jmx** option is chosen here.
JNDIName	This parameter appears only if the Measurement Mode is set to **jmx**. The JNDIName is a lookup name for connecting to the JMX connector. By default, this is *jmxrmi*. If you

Parameter	Description
	have registered the JMX connector in the RMI registry using a different lookup name, then you can change this default value to reflect the same.
JMX Remote Port	This parameter appears only if the Measurement Mode is set to **jmx**. Here, specify the port at which the **jmx** listens for requests from remote hosts. Ensure that you specify the same port that you configured in the *management.properties* file of the target JBoss server (refer to the **How to Monitor JBoss Application Server Using eG Enterprise?** chapter for more details).
JMX User, JMX Password, and Confirm Password	These parameters appear only if the Measurement Mode is set to **jmx**. If JMX requires **authentication only** (but no security), then ensure that the JMX User and JMX Password parameters are configured with the credentials of a user with *read-write* access to JMX. To know how to create this user, refer to the **How to Monitor JBoss Application Server Using eG Enterprise?** chapter. Confirm the password by retyping it in the Confirm Password text box.
SSL	This parameter appears only if the Measurement Mode is set to **War File**. Indicate **Yes** if the JBoss server is SSL-enabled.
URL	This parameter appears only if the Measurement Mode is set to **War File**. Specify the URL to be accessed to collect metrics pertaining to the JBoss server. By default, this test connects to a managed JBoss server and attempts to obtain the metrics of interest by accessing the local Mbeans of the server. Accordingly, by default, this parameter is set to *http://<JBossServerIP>:<JbossServerPort>*.
JMX Provider	This parameter appears only if the Measurement Mode is set to **jmx**. This test uses a JMX Provider to access the MBean attributes of the JBoss server and collect metrics. Specify the package name of this JMX Provider here. By default, this is set to *com.sun.jmx.remote.protocol*.
Timeout	Specify the duration (in seconds) for which this test should wait for a response from the JBoss server. If there is no response from the server beyond the configured duration, the test will timeout. By default, this is set to *240* seconds.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Creation rate	Indicates the rate of creation of beans.	EJBs/Sec	
Removal rate	Indicates the rate at which the EJBs are removed.	EJBs/Sec	

Measurement	Description	Measurement Unit	Interpretation
Message rate	Indicates the rate at which messages are handled.	Msgs/Sec	
Method ready count	Indicates the number of EJBs where the methods are currently in the ready state to service clients. A Method is a function invoked by clients on the EJB to perform a specific task.	Number	
Ready count	Indicates the number of EJBs that are currently ready to service clients.	Number	
Pooled count	Indicates the number of EJBs that are in the pool, currently.	Number	
Passive count	Indicates the number of EJB instances that have been passivated, currently. Passivation is the act of storing an EJB instance in a permanent store like a hard disk for future activation.	Number	

3.6 The JB MQ Layer

The tests associated with the **JB MQ** layer (see Figure 3.7) monitor the messaging service of the JBoss application server.

Figure 3.7: The tests associated with the JB MQ layer

The JMS API stands for Java Message Service Application Programming Interface, and is used by the JBoss server to send asynchronous business-quality messages to other applications. In the messaging world, messages are not sent directly to other applications. Instead, messages are sent to destinations. A destination is the object on the JBossMQ server that clients use to send and receive messages. There are two types of destination objects, namely, **Queues** and **Topics**. The **JbossMQQueues** test and the **JbossMQTopics** test mapped to the **JB MQ** layer monitor each of the above-mentioned destination objects, respectively.

3.6.1 Jboss MQ Queues Test

This test monitors the queues on the JBoss server. Clients that are in the point-to-point paradigm typically use queues. They expect that message sent to a queue will be received by only one other client once and only once. If multiple clients are receiving messages from a single queue, the messages will be load balanced across the receivers. Queue objects, by default, will be stored under the JNDI queue/ sub context.

Target of the test : A JBoss application server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for every queue on the JBoss application server.

Configurable parameters for the test

Parameter	Description
Test period	How often should the test be executed.
Host	The host for which the test is to be configured.
Port	The port number at which the specified host listens to.

Parameter	Description
Measurement Mode	This test can extract metrics from JBoss using either of the following mechanisms: • By deploying the **EgJBossAgent.sar** and **egjboss.war** files on the target JBoss server; • By contacting the Java runtime (JRE) of JBoss viaJMX To configure the test to usethe **EgJBossAgent.sar** and **egjboss.war** files, first, select the **War File** option. Then, refer to the **How to Monitor JBoss Application Server Using eG Enterprise?** chapter to know how to deploy these files on the target JBoss server. On the other hand, if you want the test to use JMX instead, then first, select the **jmx** option. Then, follow the procedure detailed in the **How to Monitor JBoss Application Server Using eG Enterprise?** chapter to configure the test to use **jmx**. By default, the **jmx** option is chosen here.
JNDIName	This parameter appears only if the Measurement Mode is set to **jmx**. The JNDIName is a lookup name for connecting to the JMX connector. By default, this is *jmxrmi*. If you have registered the JMX connector in the RMI registry using a different lookup name, then you can change this default value to reflect the same.
JMX Remote Port	This parameter appears only if the Measurement Mode is set to **jmx**. Here, specify the port at which the **jmx** listens for requests from remote hosts. Ensure that you specify the same port that you configured in the *management.properties* file of the target JBoss server (refer to the **How to Monitor JBoss Application Server Using eG Enterprise?** chapter for more details).
JMX User, JMX Password, and Confirm Password	These parameters appear only if the Measurement Mode is set to **jmx**. If JMX requires **authentication only** (but no security), then ensure that the JMX User and JMX Password parameters are configured with the credentials of a user with *read-write* access to JMX. To know how to create this user, refer to the **How to Monitor JBoss Application Server Using eG Enterprise?** chapter. Confirm the password by retyping it in the Confirm Password text box.
SSL	This parameter appears only if the Measurement Mode is set to **War File**. Indicate **Yes** if the JBoss server is SSL-enabled.
URL	This parameter appears only if the Measurement Mode is set to **War File**. Specify the URL to be accessed to collect metrics pertaining to the JBoss server. By default, this test connects to a managed JBoss server and attempts to obtain the metrics of interest by accessing the local Mbeans of the server. Accordingly, by default, this parameter is set to *http://<JBossServerIP>:<JbossServerPort>*.

Parameter	Description
JMX Provider	This parameter appears only if the Measurement Mode is set to **jmx**. This test uses a JMX Provider to access the MBean attributes of the JBoss server and collect metrics. Specify the package name of this JMX Provider here. By default, this is set to *com.sun.jmx.remote.protocol*.
Timeout	Specify the duration (in seconds) for which this test should wait for a response from the JBoss server. If there is no response from the server beyond the configured duration, the test will timeout. By default, this is set to *240* seconds.
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.
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.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Max queue depth	Indicates the max value of the number of messages that were in this queue since the start of the queue.	Number	
Message processing rate	Indicates the rate at which messages are being processed by this queue.	Msgs/Sec	

Measurement	Description	Measurement Unit	Interpretation
Receivers count	Indicates the number of clients that are currently configured as receivers of messages from this queue.	Msgs/Sec	
Current queue depth	Indicates the number of messages currently in this queue.	Number	A high value is indicative of server workload, or a delivery bottleneck. Use the detailed diagnosis of this measure to know the JMS ID, Priority, Time, ProducerID, and the contents of each Message in the queue.
Scheduled messages count	Indicates the number of messages in this queue that are currently scheduled to be delivered.	Number	Use the detailed diagnosis of this measure to know the JMS ID, Priority, Time, ProducerID, and the contents of each Message that is scheduled for delivery.
Queue occupied	Indicates the percentage of queue length that is occupied by messages.	Percent	A high value is a cause for concern as it could indicate a bottleneck in message delivery, which may be heavily populating the queue with messages.
Message delivered	Indicates the number of messages in this queue that have been delivered.	Number	A high value is desired for this measure.
Messages added	Indicates the number of messages that were added to this queue during the last measurement period.	Number	
Last message sent time	Indicates the time that elapsed since the last message was sent from this queue.	Secs	Ideally, the value of this measure should be low. A high value is indicative of a delay in message delivery or a message processing bottleneck.
Subscriber count	Indicates the number of subscribers who are registered with this queue.	Number	Use the detailed diagnosis of this measure to view the ID of all the subscribers registered with the queue.
Receivers count	Indicates the number of	Number	Use the detailed diagnosis of this

Measurement	Description	Measurement Unit	Interpretation
	clients that are currently configured as receivers of messages from this queue.		measure to view the ID of all the receivers configured for this queue.

3.6.2 Jboss MQ Topics Test

The JbossMQTopics test monitors the topics on the JBoss server. Topics are used in the publish-subscribe paradigm. When a client publishes a message to a topic, he expects that a copy of the message will be delivered to each client that has subscribed to the topic. However, if the client is not up, running and receiving messages from the topics, it will miss messages published to the topic. To get around this problem of missing messages, clients can start a durable subscription. This is like having a VCR record a show you cannot watch at its scheduled time so that you can see what you missed when you turn your TV back on. Similarly, messages meant for a durable subscriber are stored in the persistent cache even when the subscriber is inactive. These messages are delivered to durable subscribers when they connect to the server.

Target of the test : A JBoss application server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for every topic on the JBoss application server.

Configurable parameters for the test

Parameter	Description
Test period	How often should the test be executed.
Host	The host for which the test is to be configured.
Port	The port number at which the specified host listens to.
Measurement Mode	This test can extract metrics from JBoss using either of the following mechanisms: • By deploying the **EgJBossAgent.sar** and **egjboss.war** files on the target JBoss server; • By contacting the Java runtime (JRE) of JBoss viaJMX To configure the test to usethe **EgJBossAgent.sar** and **egjboss.war** files, first, select the **War File** option. Then, refer to the **How to Monitor JBoss Application**

Parameter	Description
	Server Using eG Enterprise? chapter to know how to deploy these files on the target JBoss server. On the other hand, if you want the test to use JMX instead, then first, select the **jmx** option. Then, follow the procedure detailed in the **How to Monitor JBoss Application Server Using eG Enterprise?** chapter to configure the test to use **jmx**. By default, the **jmx** option is chosen here.
JNDIName	This parameter appears only if the Measurement Mode is set to **jmx**. The JNDIName is a lookup name for connecting to the JMX connector. By default, this is *jmxrmi*. If you have registered the JMX connector in the RMI registry using a different lookup name, then you can change this default value to reflect the same.
JMX Remote Port	This parameter appears only if the Measurement Mode is set to **jmx**. Here, specify the port at which the **jmx** listens for requests from remote hosts. Ensure that you specify the same port that you configured in the *management.properties* file of the target JBoss server (refer to the **How to Monitor JBoss Application Server Using eG Enterprise?** chapter for more details).
JMX User, JMX Password, and Confirm Password	These parameters appear only if the Measurement Mode is set to **jmx**. If JMX requires **authentication only** (but no security), then ensure that the JMX User and JMX Password parameters are configured with the credentials of a user with *read-write* access to JMX. To know how to create this user, refer to the **How to Monitor JBoss Application Server Using eG Enterprise?** chapter. Confirm the password by retyping it in the Confirm Password text box.
SSL	This parameter appears only if the Measurement Mode is set to **War File**. Indicate **Yes** if the JBoss server is SSL-enabled.
URL	This parameter appears only if the Measurement Mode is set to **War File**. Specify the URL to be accessed to collect metrics pertaining to the JBoss server. By default, this test connects to a managed JBoss server and attempts to obtain the metrics of interest by accessing the local Mbeans of the server. Accordingly, by default, this parameter is set to *http://<JBossServerIP>:<JbossServerPort>*.
JMX Provider	This parameter appears only if the Measurement Mode is set to **jmx**. This test uses a JMX Provider to access the MBean attributes of the JBoss server and collect metrics. Specify the package name of this JMX Provider here. By default, this is set to *com.sun.jmx.remote.protocol*.
Timeout	Specify the duration (in seconds) for which this test should wait for a response from the JBoss server. If there is no response from the server beyond the configured duration, the test will timeout. By default, this is set to *240* seconds.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Max topic depth	Indicates the max value of the number of messages in the topic since the start of the queue.	Number	
Msg process rate	Indicates the rate at which messages were being processed by the topic in the last measurement period.	Msgs/Sec	
Durable messages count	Indicates the number of durable messages currently in the topic.	Number	
Non-durable messages count	Indicates the number of non-durable messages currently in the topic.	Number	
Total subjects count	Indicates the total number of subscribers to a topic.	Number	
Durable subjects count	Indicates the number of durable subscribers to the topic.	Number	If the total number of durable subscribers is high, then we can expect the total durable messages to be stored on the server also to be relatively on the higher side.
Non-durable subjects count	Indicates the number of subscribers to the topic who are currently non-durable.	Number	

About eG Innovations

eG Innovations provides intelligent performance management solutions that automate and dramatically accelerate the discovery, diagnosis, and resolution of IT performance issues in on-premises, cloud and hybrid environments. Where traditional monitoring tools often fail to provide insight into the performance drivers of business services and user experience, eG Innovations provides total performance visibility across every layer and every tier of the IT infrastructure that supports the business service chain. From desktops to applications, from servers to network and storage, from virtualization to cloud, eG Innovations helps companies proactively discover, instantly diagnose, and rapidly resolve even the most challenging performance and user experience issues.

eG Innovations is dedicated to helping businesses across the globe transform IT service delivery into a competitive advantage and a center for productivity, growth and profit. Many of the world's largest businesses use eG Enterprise to enhance IT service performance, increase operational efficiency, ensure IT effectiveness and deliver on the ROI promise of transformational IT investments across physical, virtual and cloud environments.

To learn more visit www.eginnovations.com.

Contact Us

For support queries, email support@eginnovations.com.

To contact eG Innovations sales team, email sales@eginnovations.com.