Monitoring GlassFish Server

eG Innovations Product Documentation

www.eginnovations.com

eG
Total Performance Visibility

Table of Contents

Table of Figures

Chapter 1: Introduction

Oracle GlassFish Server is the world's first implementation of the Java Platform, Enterprise Edition (Java EE) 6 specification. Built using the GlassFish Server Open Source Edition, Oracle GlassFish Server delivers a flexible, lightweight and production-ready Java EE 6 platform.

Oracle GlassFish Server is based on the Java EE 6 Reference Implementation and is the first application server to support the full Java EE 6 platform and the new Java EE 6 Web Profile, which is designed specifically for Web applications. Oracle GlassFish Server delivers a new, efficient model for developing and deploying production-level applications. Developers can begin with just the Web Profile and grow to the full Java EE 6 platform when needed. Because Oracle GlassFish Server runs on the OSGi runtime, modular features can be added as necessary. Developers can also create hybrid applications - using OSGi services from Java EE Applications or using Java EE services from OSGi services. It also keeps the footprint as small as possible by loading only modules required to service deployed applications, improving startup time and reducing resource utilization.

Because it is swift and resource-efficient, the Oracle GlassFish server is the preferred middle-ware for mission-critical IT infrastructures where performance and prompt delivery of services to end-users is of utmost importance. In such environments, if the service level guarantees of the GlassFish server are not fulfilled, it is sure to result in dissatisfied users, severe penalties, and loss of revenue and reputation. You hence need to continuously monitor the performance of the GlassFish server, so that service level slippages are minimized and user satisfaction with the overlying business services is maximized. This is where eG Enterprise helps administrators to fulfill their duty in this regard.

Chapter 2: How to Monitor the Oracle GlassFish Enterprise Server Using eG Enterprise?

eG Enterprise monitors the GlassFish server in both agent-based or agentless manners. All that is required for this is a single eG agent on any remote Windows host in the environment. To enable the eG agent to pullout metrics on the performance of the GlassFish server, a set of pre-requisites should be fulfilled. These requirements have been discussed in the section below.

2.1 Pre-requisites for Monitoring Oracle GlassFish Enterprise Server

To enable the eG agent to collect metrics on health of key components of the application server such as the servlets, EJBs, connection pools, caches, the JVM, and more!, you need to make sure that the following pre-requisites are fulfilled:

1. The eG agent should be configured to use **JMX** to connect to the **JRE** of the GlassFish server and collect the metrics of interest. By default, **JMX** support is enabled for GlassFish. If, for some reason, **JMX** is disabled for the GlassFish server in your environment, then, follow the steps below to enable it:

   - In an editor, open the domain.xml file in the <glassfish_install_dir>\Sun\AppServer\domains\domain1\config directory and look for the code block shown below:

   ```
   <jmx-connector accept-all="false" address="0.0.0.0" auth-realm-name="admin-realm" enabled="false" name="system" port="8686" protocol="rmi_jrmp" security-enabled="false">
   ```

   - If JMX is disabled for the GlassFish server, then the enabled parameter in the code block above will be set to **false**. To enable JMX, set the enabled parameter to **true** as shown below:

   ```
   <jmx-connector accept-all="false" address="0.0.0.0" auth-realm-name="admin-realm" enabled="true" name="system" port="8686" protocol="rmi_jrmp" security-enabled="false">
   ```

   - Against the port parameter, configure the port at which JMX listens. The default JMX port is *8686*.

   - Finally, save the file.

2. Since JMX on GlassFish requires authentication by default, every test executed by the eG agent

should be configured with the credentials of an *Administrator* to the GlassFish server. If you prefer not to expose the administrator credentials owing to security compulsions, then configure the test with the credentials of any valid user who has the right to use JMX. To know the name of such a user, do the following:

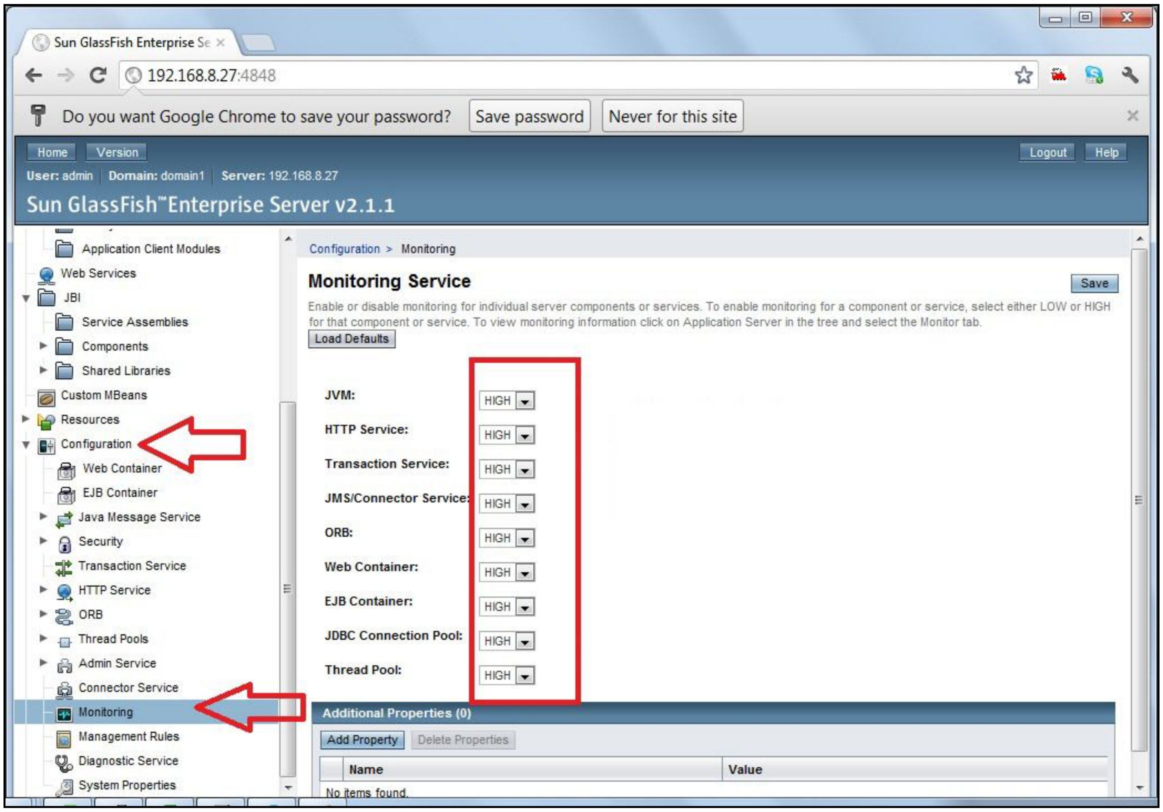
- In an editor, open the domain.xml file in the <glassfish_ install_ dir>\Sun\AppServer\domains\domain1\config directory and look for the code block shown below:

```
<jmx- connector accept- all="false" address="0.0.0.0" auth- realm- name="admin- realm"
enabled="true" name="system" port="8686" protocol="rmi_ jrmp" security-
enabled="false">
```

- The user name specified against the auth-realm-name parameter in the code block above can be configured as the user, and the password of that user can be specified against JMX PASSWORD.

3. You should enable the **Monitoring service** of the GlassFish server. To achieve this, do the following:

   - Connect to the GlassFish server using the URL: *http://<GlassFishServer_ IP>:<GlassFishServer_Port>/*
   - Login to the server as administrator.
   - Figure 2.1 will then appear. Expand the Configuration node in the tree-structure in the left panel of Figure 2.1, and select the **Monitoring sub-node** within.



Do you want Google Chrome to save your password?

User: admin Domain: domain1 Server: 192.168.8.27

Sun GlassFish™Enterprise Server v2.1.1

Monitoring Service

Enable or disable monitoring for individual server components or services. To enable monitoring for a component or service, select either LOW or HIGH for that component or service. To view monitoring information click on Application Server in the tree and select the Monitor tab.

JVM: HIGH

HTTP Service: HIGH

Transaction Service: HIGH

JMS/Connector Service: HIGH

ORB: HIGH

Web Container: HIGH

EJB Container: HIGH

JDBC Connection Pool: HIGH

Thread Pool: HIGH

Additional Properties (0)

Name	Value
No items found.	

Figure 2.1: Enabling the Monitoring Service

- Set the monitoring-level for all the server components listed in the right panel to high.
- Finally, click the **Save** button in the right-panel to save the changes.

Once the above-said pre-requsites are fulfilled, start monitoring the GlassFish server using the eG Enterprise. The broad steps for monitoring the server using eG Enterprise are as follows:

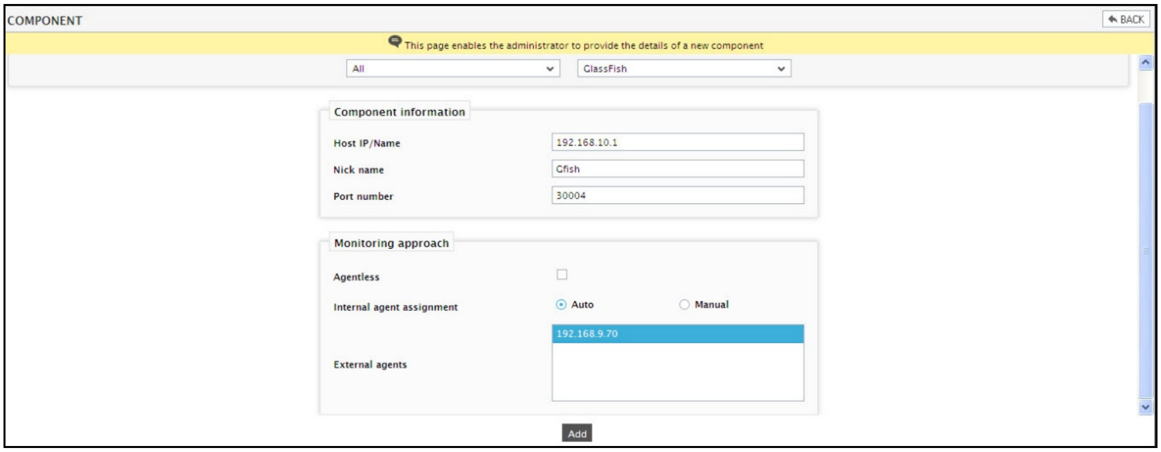
- Managing the Groupwise Internet Application (GWIA)
- Configuring the tests

These steps have been discussed in following sections.

2.2 Managing the GlassFish server

The eG Enterprise cannot automatically discover the GlassFish server. This implies that you need to manually add the component for monitoring. Remember that the eG Enterprise automatically manages the components that are added manually. To manage a GlassFish server component, do the following:

1. Log into the eG administrative interface.
2. Follow the Components -> Add/Modify menu sequence in the **Infrastructure** tile of the **Admin** menu.
3. In the **COMPONENT** page that appears next, select *GlassFish* as the **Component type**. Then, click the **Add New Component** button. This will invoke Figure 2.2.



This page enables the administrator to provide the details of a new component

All | GlassFish

Component information

Host IP/Name: 192.168.10.1

Nick name: Gfish

Port number: 30004

Monitoring approach

Agentless: □

Internal agent assignment: ● Auto ○ Manual

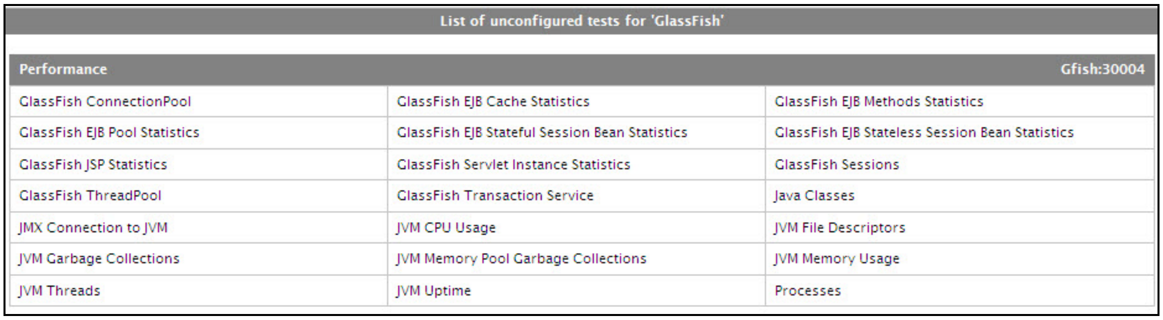
External agents: 192.168.9.70

Figure 2.2: Adding the GlassFish server

4. Specify the **Host IP,** the **Nick name** and **Port Number** for the GlassFish server in 2.2. Then, click the **Add** button to register the changes.

2.3 Configuring the tests

1. When you attempt to sign out of eG administrative interface, a list of unconfigured tests will appear as shown in Figure 2.3. This list reveals the unconfigured tests requiring manual configuration.



List of unconfigured tests for 'GlassFish'

Performance		Gfish:30004
GlassFish ConnectionPool	GlassFish EJB Cache Statistics	GlassFish EJB Methods Statistics
GlassFish EJB Pool Statistics	GlassFish EJB Stateful Session Bean Statistics	GlassFish EJB Stateless Session Bean Statistics
GlassFish JSP Statistics	GlassFish Servlet Instance Statistics	GlassFish Sessions
GlassFish ThreadPool	GlassFish Transaction Service	Java Classes
JMX Connection to JVM	JVM CPU Usage	JVM File Descriptors
JVM Garbage Collections	JVM Memory Pool Garbage Collections	JVM Memory Usage
JVM Threads	JVM Uptime	Processes

Figure 2.3: List of tests to be configured for the GlassFish server

2. To know how to configure the tests, refer to **Monitoring the Oracle GlassFish Enterprise Server**.
3. Next, when you signout of the administrative interface, you will be prompted to configure the **Processes** test. To know how to configure this test, refer to the *Monitoring Unix and Windows Servers* document.
4. Once the **Processes** test is configured, signout of the eG administrative interface.

Chapter 3: Monitoring the Oracle GlassFish Enterprise Server

eG Enterprise provides a specialized GlassFish monitoring model that monitors the health of the Oracle GlassFish server inside-out and promptly captures and reports abnormalities.

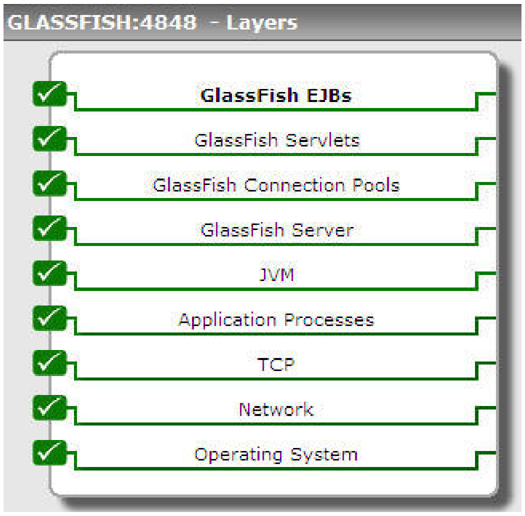


Figure 3.1: The layer model of the GlassFish server

Every layer of Figure 1 above is mapped to a variety of tests; each of these tests can be configured to use *agent-based* or *agentless* methodologies to report on the health of key components of the application server such as the servlets, EJBs, connection pools, caches, the JVM, and more!

Once the pre-requisites discussed in Section **2.1** are fulfilled, the eG agent will execute the tests and collect the desired performance metrics from the GlassFish server. Using the statistics reported by these tests, administrators can find quick and accurate answers for the following performance queries:

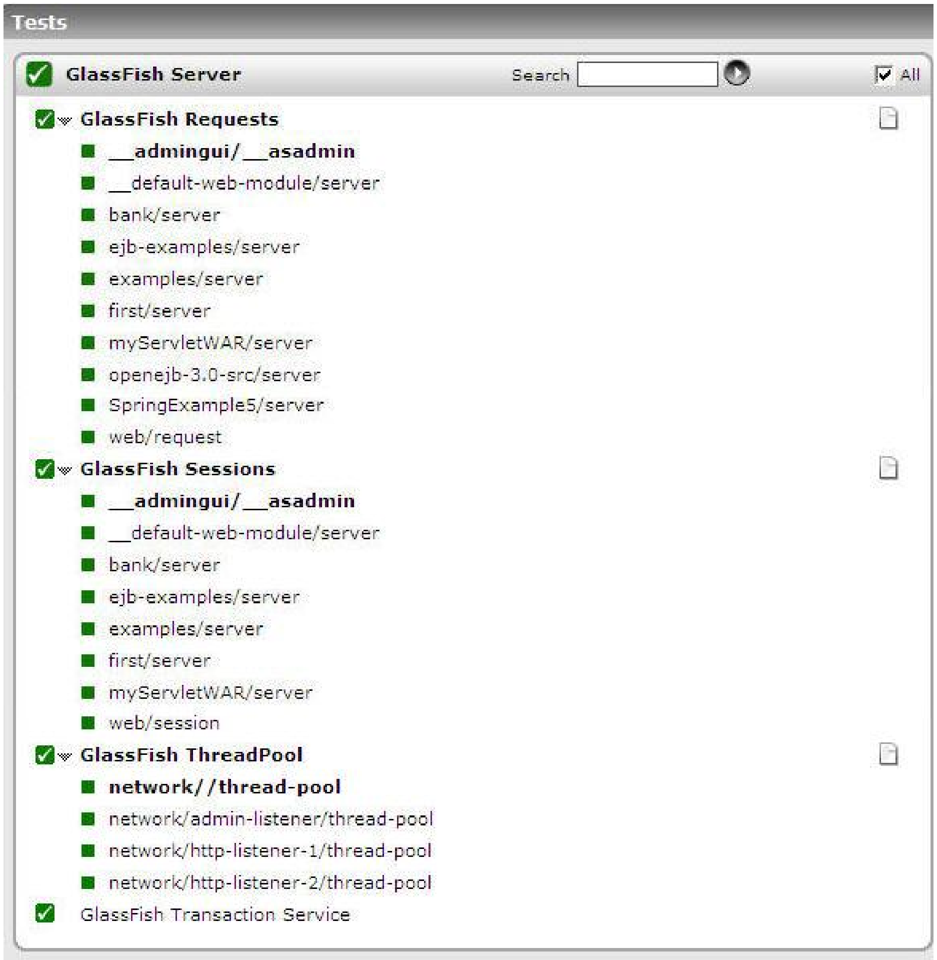
- Is any application deployed on the server experiencing a request overload? If so. which application is this?
- Is any application processing requests slowly? Which application is this and what is causing the processing delay - is it because of poorly responsive servlets? is it because of JSP errors? is it owing to EJB caches running out of beans or EJB-related errors? is it because of one/more inefficient EJB methods? Which servlet/JSP/EJB cache/EJB pool/EJB method is contributing to this slowdown?
- Did any application send out an error response recently?

- Which application is overloaded with sessions?
- Has any application rejected any sessions?
- Are all thread pools utilizing threads optimally? Are any thread pools guilty of ineffective thread usage?
- Do the thread pools have adequate threads to handle the request load?
- Are transaction rollbacks kept at a minimum?
- Are requests to any connection pool waiting a long time for connections? Are too many such requests in waiting? Is it because there are not enough connections in the pool to service the requests?
- Has a potential connection leak been detected in any pool?

The bottom four layers of the layer model have been dealt with in the *Monitoring Unix and Windows Servers* document. The **JVM** layer has already been discussed elaborately in the *Monitoring Java Applications* document. The sections that follow will therefore focus on the top 4 layers of layer model only.

3.1 The GlassFish Server Layer

Using the tests mapped to this layer, administrators can measure the session, request, and transaction load to the GlassFish server and determine how well the server handles the load.



Tests

☑ GlassFish Server — Search ☑ All

- ☑ GlassFish Requests
  - __admingui/__asadmin
  - __default-web-module/server
  - bank/server
  - ejb-examples/server
  - examples/server
  - first/server
  - myServletWAR/server
  - openejb-3.0-src/server
  - SpringExample5/server
  - web/request
- ☑ GlassFish Sessions
  - __admingui/__asadmin
  - __default-web-module/server
  - bank/server
  - ejb-examples/server
  - examples/server
  - first/server
  - myServletWAR/server
  - web/session
- ☑ GlassFish ThreadPool
  - network//thread-pool
  - network/admin-listener/thread-pool
  - network/http-listener-1/thread-pool
  - network/http-listener-2/thread-pool
- ☑ GlassFish Transaction Service

Figure 3.2: The tests mapped to the GlassFish Server layer

3.1.1 GlassFish Requests Test

The true test for the efficiency of a web application server, such as the GlassFish Enterprise server, is the speed with which it processes requests to its web applications. The **GlassFish Requests** test enables administrators to judge the efficiency of GlassFish by monitoring the requests to each web application deployed on it, and proactively alerting them to current or potential bottlenecks in request processing. This way, overloaded applications and the least responsive applications can be isolated. In the process, the test also promptly captures errors (if any) that the server may have experienced during request processing.

Target of the test : A GlassFish server

Agent deploying the test : An internal/remote agent

Outputs of the test : One set of results for each web application deployed on the GlassFish Enterprise server being monitored.

Configurable parameters for the test

Parameters	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
JMX Remote Port	To collect metrics from a GlassFish Enterprise server, the eG agent monitoring that server should be configured to use **JMX** to connect to the JRE used by the server and pull out the metrics of interest. By default, **JMX** support is enabled for the JRE used by the GlassFish Enterprise Server. The JMX connector listens on port 8686 by default. Therefore, type 8686 as the JMX Remote Port. If JMX listens on a different port in your environment, then specify the same here. To know the port at which JMX listens, open the **domain.xml** file in the <GLASSFISH_INSTALL_DIR>\Sun\AppServer\domains\domain1\config directory and look for the code block shown below: `<jmx-connector accept-all="false" address="0.0.0.0" auth-realm-name="admin-realm" enabled="true" name="system" port="8686" protocol="rmi_jrmp" security-enabled="false">` The port number specified against the Port parameter in the code block above refers to the JMX Remote Port.
JMX User, JMX Password, and Confirm Password	By default, the JMX connector on the GlassFish Enterprise Server requires authentication. To enable the eG agent to use JMX therefore, you need to configure the agent with the credentials of a user who is authorized to use JMX. Typically, the GlassFish administrator has the right to use JMX. You can hence configure the JMX User and JMX Password parameters with the credentials of the *administrator*. However, if you prefer not to expose the credentials of an administrator owing to security considerations, you can use the credentials of any other user with access rights to JMX. To know the name of such a user, open the **domain.xml** file in the <GLASSFISH_INSTALL_DIR>\Sun\AppServer\domains\domain1\config directory and look for the code block shown below: `<jmx-connector accept-all="false" address="0.0.0.0" auth-realm-name="admin-realm" enabled="true" name="system" port="8686" protocol="rmi_jrmp" security-enabled="false">` The user name specified against the auth-realm-name parameter in the code block above can be configured as the JMX User, and the Password of that user can be

Parameters	Description
	specified against JMX Password. Confirm the JMX Password you specify by retyping that password in the Confirm Password text box.
JNDIName	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 Provider	This test uses a JMX Provider to access the MBean attributes of the GlassFish Enterprise 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 GlassFish Enterprise 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
Requests processed	Indicates the number of requests to this application that were processed during the last measurement period.	Number	This measure is a good indicator of the load on each application. Compare the value of this measure across applications to identify which application is experiencing very high load.
Average processing time	Indicates the time taken by this application to process the requests to it.	Secs	A low value is desired for this measure. A consistent rise in the value of this measure could indicate a processing bottleneck, which in turn may affect application performance. Compare the value of this measure across applications to know which application is the least responsive to user requests.
Maximum time	Indicates the maximum time taken by this application to process requests.	Secs	
Errors	Indicates the number of	Number	This count includes the following:

Measurement	Description	Measurement Unit	Interpretation
	error responses that were sent by this application during the last measurement period.		• The number of requests for which responses with response code that is greater than or equal to 400 were sent; • The number of requests for which a response could not be sent at all. Ideally, the value of this measure should be 0.
Error	Indicates the percentage of error responses that were sent by this application during the last measurement period	Percent	Ideally, the value of this measure should be 0.

3.1.2 GlassFish Sessions Test

To understand how much load is imposed by user sessions on each of the web applications deployed on the GlassFish server, and to determine the nature (eg., persisted sessions, rejected sessions, etc.) and the current state of these sessions (whether activated, passivated, created, etc.), use the **GlassFish Sessions** test. This way, the most popular web applications on the server can be isolated, and those applications that reject sessions too often can be identified.

Target of the test : A GlassFish server

Agent deploying the test : An internal/remote agent

Outputs of the test : One set of results for each web application deployed on the GlassFish Enterprise server being monitored.

Configurable parameters for the test

Parameters	Description
Test Period	How often should the test be executed.
Host	The host for which the test is to be configured.

Parameters	Description
Port	The port number at which the specified host listens
JMX Remote Port	To collect metrics from a GlassFish Enterprise server, the eG agent monitoring that server should be configured to use **JMX** to connect to the JRE used by the server and pull out the metrics of interest. By default, **JMX** support is enabled for the JRE used by the GlassFish Enterprise Server. The JMX connector listens on port 8686 by default. Therefore, type 8686 as the JMX Remote Port. If JMX listens on a different port in your environment, then specify the same here. To know the port at which JMX listens, open the **domain.xml** file in the <GLASSFISH_INSTALL_DIR>\Sun\AppServer\domains\domain1\config directory and look for the code block shown below: `<jmx-connector accept-all="false" address="0.0.0.0" auth-realm-name="admin-realm" enabled="true" name="system" port="8686" protocol="rmi_jrmp" security-enabled="false">` The port number specified against the Port parameter in the code block above refers to the JMX Remote Port.
JMX User, JMX Password, and Confirm Password	By default, the JMX connector on the GlassFish Enterprise Server requires authentication. To enable the eG agent to use JMX therefore, you need to configure the agent with the credentials of a user who is authorized to use JMX. Typically, the GlassFish administrator has the right to use JMX. You can hence configure the JMX User and JMX Password parameters with the credentials of the *administrator*. However, if you prefer not to expose the credentials of an administrator owing to security considerations, you can use the credentials of any other user with access rights to JMX. To know the name of such a user, open the **domain.xml** file in the <GLASSFISH_INSTALL_DIR>\Sun\AppServer\domains\domain1\config directory and look for the code block shown below: `<jmx-connector accept-all="false" address="0.0.0.0" auth-realm-name="admin-realm" enabled="true" name="system" port="8686" protocol="rmi_jrmp" security-enabled="false">` The user name specified against the auth-realm-name parameter in the code block above can be configured as the JMX User, and the Password of that user can be specified against JMX Password. Confirm the JMX Password you specify by retyping that password in the Confirm Password text box.
JNDIName	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 Provider	This test uses a JMX Provider to access the MBean attributes of the GlassFish Enterprise server and collect metrics. Specify the package name of this JMX Provider

Parameters	Description
	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 GlassFish Enterprise 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 active sessions	Indicates the number of sessions that are currently active on this application.	Number	This is a good indicator of the current workload of the application. You can compare the value of this measure across applications to know which applications is the most popular in terms of the number of user sessions open on it.
Activated sessions	Indicates the number of sessions to this application that were activated during the last measurement period.	Number	
Passivated sessions	Indicates the number of sessions to this application that were passivated during the last measurement period.	Number	
Expired sessions	Indicates the number of sessions to this application that expired during the last measurement period.	Number	This measure indicates the number of sessions that remained inactive on the application for a duration in excess of the session-timeout value configured at the web application level. If the value of this measure is unreasonably high, you may want to increase the session-timeout value, so as to minimize the frequency of session expiry. If the session-timeout value is set to 0 or less, then sessions

Measurement	Description	Measurement Unit	Interpretation
			to that web application will never expire.
Persisted sessions	Indicates the number of sessions to this application that persisted during the last measurement period.	Number	GlassFish Server provides high availability session persistence through failover of HTTP session data and stateful session bean (SFSB) session data. Failover means that in the event of a server instance or hardware failure, another server instance in a cluster takes over a distributed session.
Rejected sessions	Indicates the number of sessions to this application that were rejected during the last measurement period.	Number	A low value is desired for this measure.
Sessions created	Indicates the number of sessions to this application that were created during the last measurement period.	Number	A session manager automatically creates new session objects whenever a new session starts.

3.1.3 GlassFish Thread Pool Test

The Virtual Machine for the Java platform (Java Virtual Machine) or JVM machine) can support many threads of execution simultaneously. To help performance, GlassFish Server maintains one or more thread pools. It is possible to assign specific thread pools to connector modules, to network listeners, or to the Object Request Broker (ORB).

One thread pool can serve multiple connector modules and enterprise beans. Request threads handle user requests for application components. When GlassFish Server receives a request, it assigns the request to a free thread from the thread pool. The thread executes the client's requests and returns results.

Administrators will have to continuously observe the request load on their applications, periodically check how the thread pool services these requests, and accordingly decide whether/not to resize the thread pool commensurate to the current and anticipated load. The **GlassFish Thread Pool** test provides administrators with the load and thread pool usage insights they require to take these crucial sizing decisions.

Target of the test : A GlassFish server

Agent deploying the test : An internal/remote agent

Outputs of the test : One set of results for each thread pool configured on the GlassFish Enterprise server being monitored.

Configurable parameters for the test

Parameters	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
JMX Remote Port	To collect metrics from a GlassFish Enterprise server, the eG agent monitoring that server should be configured to use **JMX** to connect to the JRE used by the server and pull out the metrics of interest. By default, **JMX** support is enabled for the JRE used by the GlassFish Enterprise Server. The JMX connector listens on port 8686 by default. Therefore, type 8686 as the JMX Remote Port. If JMX listens on a different port in your environment, then specify the same here. To know the port at which JMX listens, open the **domain.xml** file in the <GLASSFISH_INSTALL_ DIR>\Sun\AppServer\domains\domain1\config directory and look for the code block shown below: `<jmx-connector accept-all="false" address="0.0.0.0" auth-realm-name="admin-realm" enabled="true" name="system" port="8686" protocol="rmi_jrmp" security-enabled="false">` The port number specified against the Port parameter in the code block above refers to the JMX Remote Port.
JMX User, JMX Password, and Confirm Password	By default, the JMX connector on the GlassFish Enterprise Server requires authentication. To enable the eG agent to use JMX therefore, you need to configure the agent with the credentials of a user who is authorized to use JMX. Typically, the GlassFish administrator has the right to use JMX. You can hence configure the JMX User and JMX Password parameters with the credentials of the *administrator*. However, if you prefer not to expose the credentials of an administrator owing to security considerations, you can use the credentials of any other user with access rights to JMX. To know the name of such a user, open the **domain.xml** file in the <GLASSFISH_INSTALL_DIR>\Sun\AppServer\domains\domain1\config directory

Parameters	Description
	and look for the code block shown below: `<jmx-connector accept-all="false" address="0.0.0.0" auth-realm-name="admin-realm" enabled="true" name="system" port="8686" protocol="rmi_jrmp" security-enabled="false">` The user name specified against the auth-realm-name parameter in the code block above can be configured as the JMX User, and the Password of that user can be specified against JMX Password. Confirm the JMX Password you specify by retyping that password in the Confirm Password text box.
JNDIName	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 Provider	This test uses a JMX Provider to access the MBean attributes of the GlassFish Enterprise 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 GlassFish Enterprise 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
Core threads	Indicates the number of core threads currently in this pool.	Number	
Current threads	Indicates the current number of threads in this pool.	Number	
Current busy threads	Indicates the number of threads in this pool that are currently processing requests.	Number	A high value for this measure is indicative of a high load on this applications deployed on the GlassFish application server. This measure is also useful for determining usage trends. For example, it can show the time of day and the day of the week in which you

Measurement	Description	Measurement Unit	Interpretation
			usually reach peak thread count. In addition, the creation of too many threads can result in out of memory errors or thrashing. By watching this metric, you can reduce excessive memory consumption before it's too late.
Maximum threads	Indicates the maximum number of threads allowed in this thread pool.	Number	This measure reports the value of the Max Thread Pool Size parameter of the GlassFish Enterprise server. This parameter specifies the maximum number of simultaneous requests the server can handle. The default value is 5. When the server has reached the limit or request threads, it defers processing new requests until the number of active requests drops below the maximum amount. Increasing this value will reduce HTTP response latency times.
Total executed tasks	Indicates the total number of tasks executed by the threads in this pool during the last measurement period.	Number	This measure is a good indicator of how busy the pool was during the last measurement period.
Thread busy	Indicates the percentage of threads in this pool that are currently in use.	Percent	This measure is computed using the following formula: (Current busy threads / Maximum threads)*100 A high value is indicative of a busy pool. A value close to 100% indicates excessive utilization of threads in a pool. If the value of this measure grows closer to 100% over time, it indicates that the pool is rapidly running out of threads to service the request load. You may then have to

Measurement	Description	Measurement Unit	Interpretation
			fine-tune the thread pool size using the Max Thread Pool Size parameter and the Min Thread Pool Size parameter. The Max Thread Pool Size parameter specifies the maximum number of simultaneous requests the server can handle. The default value is 5. When the server has reached the limit or request threads, it defers processing new requests until the number of active requests drops below the maximum amount. Increasing this value will reduce HTTP response latency times. In practice, clients frequently connect to the server and then do not complete their requests. In these cases, the server waits a length of time specified by the Timeout parameter. Also, some sites do heavyweight transactions that take minutes to complete. Both of these factors add to the maximum simultaneous requests that are required. If your site is processing many requests that take many seconds, you might need to increase the number of maximum simultaneous requests. Adjust the thread count value based on your load and the length of time for an average request. In general, increase this number if you have idle CPU time and requests that are pending; decrease it if the CPU becomes overloaded. Suitable Request Max Thread Pool Size values range from 100 to 500,

Measurement	Description	Measurement Unit	Interpretation
			depending on the load. If your system has extra CPU cycles, keep incrementally increasing thread count and monitor performance after each incremental increase. When performance saturates (stops improving), then stop increasing thread count. The Min Thread Pool Size property specifies the minimum number of threads the server initiates upon startup. The default value is 2. Min Thread Pool Size represents a hard limit for the maximum number of active threads that can run simultaneously, which can become a bottleneck for performance. Specifying the same value for minimum and maximum threads allows GlassFish Server to use a slightly more optimized thread pool. This configuration should be considered unless the load on the server varies quite significantly.
Pool utilization	Indicates the percentage of threads in the pool that are active.	Percent	This measure is computed as a ratio of the value of the Current threads measure and the Maximum threads measure.

3.1.4 GlassFish Transaction Service Test

A transaction is a series of discreet actions in an application that must all complete successfully. By enclosing one or more actions in an indivisible unit of work, a transaction ensures data integrity and consistency.

If all these actions complete successfully, the transaction is committed. If any action fails, all changes from the preceding steps are rolled back. This type of event is called a rollback. A normal transaction ends in either a committed state or a rolled back state.

The transaction manager makes it possible to commit and roll back distributed transactions.

By monitoring the transaction manager, you can determine the transaction load on the GlassFish Enterprise server, detect transaction rollbacks, and receive regular updates on transaction status. The Transaction Service test does all the above.

Target of the test : A GlassFish server

Agent deploying the test : An internal/remote agent

Outputs of the test : One set of results for the GlassFish Enterprise server being monitored.

Configurable parameters for the test

Parameters	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
JMX Remote Port	To collect metrics from a GlassFish Enterprise server, the eG agent monitoring that server should be configured to use **JMX** to connect to the JRE used by the server and pull out the metrics of interest. By default, **JMX** support is enabled for the JRE used by the GlassFish Enterprise Server. The JMX connector listens on port 8686 by default. Therefore, type 8686 as the JMX Remote Port. If JMX listens on a different port in your environment, then specify the same here. To know the port at which JMX listens, open the **domain.xml** file in the <GLASSFISH_INSTALL_DIR>\Sun\AppServer\domains\domain1\config directory and look for the code block shown below: `<jmx-connector accept-all="false" address="0.0.0.0" auth-realm-name="admin-realm" enabled="true" name="system" port="8686" protocol="rmi_jrmp" security-enabled="false">` The port number specified against the Port parameter in the code block above refers to the JMX Remote Port.
JMX User, JMX Password, and Confirm Password	By default, the JMX connector on the GlassFish Enterprise Server requires authentication. To enable the eG agent to use JMX therefore, you need to configure the agent with the credentials of a user who is authorized to use JMX. Typically, the

Parameters	Description
	GlassFish administrator has the right to use JMX. You can hence configure the JMX User and JMX Password parameters with the credentials of the *administrator*. However, if you prefer not to expose the credentials of an administrator owing to security considerations, you can use the credentials of any other user with access rights to JMX. To know the name of such a user, open the **domain.xml** file in the <GLASSFISH_INSTALL_DIR>\Sun\AppServer\domains\domain1\config directory and look for the code block shown below: `<jmx-connector accept-all="false" address="0.0.0.0" auth-realm-name="admin-realm" enabled="true" name="system" port="8686" protocol="rmi_jrmp" security-enabled="false">` The user name specified against the auth-realm-name parameter in the code block above can be configured as the JMX User, and the Password of that user can be specified against JMX Password. Confirm the JMX Password you specify by retyping that password in the Confirm Password text box.
JNDIName	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 Provider	This test uses a JMX Provider to access the MBean attributes of the GlassFish Enterprise 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 GlassFish Enterprise 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
Active transactions	Indicates the number of transactions that were active on the server during the last measurement period.	Number	This is a good indicator of the transaction load on the server.
Rollback transactions	Indicates the number of transactions that were rolled back during the last measurement period.	Number	Rollbacks are resource-intensive operations. Ideally therefore, the value of this measure should be low.

<table>
<tr><th>Measurement</th><th>Description</th><th>Measurement Unit</th><th>Interpretation</th></tr>
<tr><td>Committed transactions</td><td>Indicates the number of transactions that were committed during the last measurement period.</td><td>Number</td><td></td></tr>
<tr><td>Transaction state</td><td>Indicates whether the transactions are currently in a frozen state or not.</td><td></td><td>If a transaction that you intend to rollback is active, then you will have to freeze that transaction before attempting the rollback. Freezing a transaction avoids the possibility of that transaction completing before the rollback request is issued.
This measure reports the value <b>True</b> if the transaction is currently frozen and reports the value <b>False</b> if the transaction is currently active.
The numeric values that correspond to the measure values described above are listed in the table below:
<table><tr><th>Measure Value</th><th>Numeric Value</th></tr><tr><td>True</td><td>1</td></tr><tr><td>False</td><td>0</td></tr></table><b>Note:</b>
This measure reports the <b>Measure Value</b>s listed in the table above to indicate the state of transactions. However, in the graph of this measure, the transaction state is indicated using only the <b>Numeric Value</b>s listed in the above table.</td></tr>
</table>

3.1.5 Web Service Test

A web service is a collection of open protocols and standards used for exchanging data between applications or systems. Software applications written in various programming languages and running on various platforms can use web services to exchange data over computer networks like the Internet in a manner similar to inter-process communication on a single computer. A complete web service is, therefore, any service that:

- Is available over the Internet or private (intranet) networks
- Uses a standardized XML messaging system
- Is not tied to any one operating system or programming language
- Is self-describing via a common XML grammar
- Is discoverable via a simple find mechanism

The basic web services platform is XML + HTTP. All the standard web services work using the following components:

- SOAP (Simple Object Access Protocol)
- UDDI (Universal Description, Discovery and Integration)
- WSDL (Web Services Description Language)

A web service enables communication among various applications by using open standards such as HTML, XML, WSDL, and SOAP. A web service takes the help of the following:

- XML to tag the data
- SOAP to transfer a message
- WSDL to describe the availability of service.

The following are the major uses of the Web Services:

- **Reusable application-components**: Often applications need repeated access to application-components like currency conversion, weather reports, or even language translation. In such cases, the web services can be used to offer the application-components as services with ease.
- **Connect existing software**: Web services can help to solve the interoperability problem by giving different applications a way to link their data. With Web services you can exchange data between different applications and different platforms. Any application can have a Web Service component. Web Services can be created regardless of programming language.

In certain environments, administrators are required to keep an eye on the web services that offer repeated access to the application-components i.e., operations so that the work load on the users using those applicaiton components can be minimized. If for some reason the web service takes too long to respond or is unavailable to cater to the needs of the users, then the users will be deprived of access to the application- components involved in that particular web service. To avoid such inconvenience caused to the users, administrators are required to continuously monitor the web services. The **Web Service** test helps administrators to perform this task perfectly. By continuously monitoring each operation i.e., application component of a web service that is offered, using the SOAP commands, this test helps administrators identify the availability, response time and response code of the web service and quickly figure out discrepancies if any web service is deemed unavailable. This way, the web services can be kept available round the clock thus helping the users perform their tasks without any difficulty.

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 *WebService:Operation* i.e., application-component performed on the target server that is being monitored.

Configurable parameters for the test

Parameters	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
WSDL URL	This test emulates a user accessing a specific web service(s) on the target server to determine the availability and responsiveness of the server. to enable this emulation, you need to configure the test with the url of the web service that it should access. specify this url against the WSDL URL parameter. if required, you can even configure multiple WSDL URLs - one each for every web service that the test should attempt to access. if each WSDL URL configured requires special permissions for logging in, then, you need to configure the test with separate credentials for logging into every WSDL URL. likewise, you need to provide instructions to the test on how to validate the content returned by every WSDL URL, and also set an encoding format for each wsdl url. to enable administrators to easily configure the above per WSDL URL, eg enterprise provides a special interface. to access this interface, click on the encircled '+' button alongside theURL text box in the test configuration page. alternatively, you can even click on the encircled '+' button adjacent to the WSDL URL parameter in the test configuration page. to know how to use this special interface, refer to Section

Parameters	Description
	3.1.5.1.
Operations	Once the WSDL URL(s) are specified, the operations that are offered by the web services and those that are to be monitored have to be configured. To select the required operations for monitoring, eG Enterprise provides a special interface. To access this interface, click on the encircled '+' button alongside the Operations text box in the test configuration page. Alternatively, you can even click on the encircled '+' button adjacent to the Operations parameter in the test configuration page. To know how to use this special interface, refer to Section **3.1.5.2**.
Timeout	Specify the duration (in seconds) for which this test should wait for a response from the server. If there is no response from the server beyond the configured duration, the test will timeout. By default, this is set to 30 seconds.
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
WSDL URL availability	Indicates whether the web service was able to respond successfully to the query made by the test.	Percent	Availability failures could be caused by several factors such as the web service process(es) being down, the web service being misconfigured, a network failure, etc. Temporary unavailability may also occur if the web service is overloaded. Availability is determined based on the response code returned by the service. A response code between 200 to 300

<table>
<tr><th>Measurement</th><th>Description</th><th>Measurement Unit</th><th>Interpretation</th></tr>
<tr><td></td><td></td><td></td><td>indicates that the service is available.</td></tr>
<tr><td>WSDL response time</td><td>Indicates the time taken by the eG agent to get the configured web service.</td><td>Secs</td><td>Response time being high denotes a problem. Poor response times may be due to the service being overloaded or misconfigured. If the URL accessed involves the generation of dynamic content by the service, backend problems (e.g., an overload at the application server or a database failure) can also result in an increase in response time.</td></tr>
<tr><td>Port status</td><td>Indicates whether/not the port of the web server is reachable.</td><td></td><td>The values reported by this measure and the corresponding numeric equivalents are listed in the table below:
<table>
<tr><th>Measure Values</th><th>Numeric Values</th></tr>
<tr><td>Yes</td><td>1</td></tr>
<tr><td>No</td><td>0</td></tr>
</table>
<b>Note:</b>
By default, this measure reports the above-mentioned <b>Measure Value</b>s to indicate whether the server has been rebooted or not. In the graph of this measure however, the <b>Measure Value</b>s are represented using the numeric equivalents only.</td></tr>
<tr><td>TCP connection availability</td><td>Indicates whether the test managed to establish a TCP connection to the server.</td><td>Percent</td><td>Failure to establish a TCP connection may imply that either the web server process is not up, or that the process is not operating correctly. In some cases of extreme overload, the failure to establish a TCP connection may be a transient condition. As the load subsides, the server may start</td></tr>
</table>

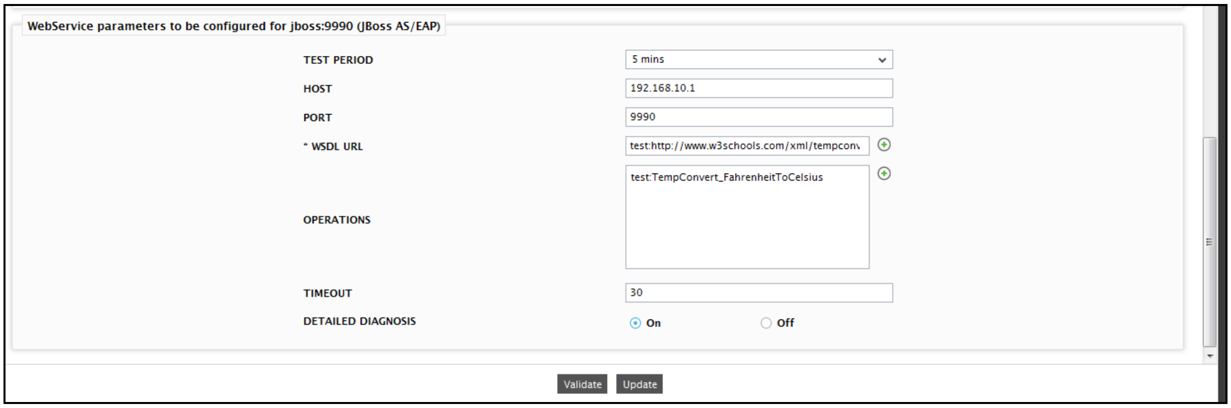
Measurement	Description	Measurement Unit	Interpretation
			functioning properly again.
TCP connect time	This measure quantifies the time for establishing a TCP connection to the web server host.	Secs	Typically, the TCP connection establishment must be very small (of the order of a few milliseconds). Since TCP connection establishment is handled at the OS-level, rather than by the application, an increase in this value signifies a system-level bottleneck on the host that supports the web server.
Server response time	Indicates the time period between when the connection was established and when the web server sent back a response header to the client.	Secs	While the total response time may depend on several factors, this measure is typically, a very good indicator of a server bottleneck (e.g., because all the available server threads or processes are in use).
Response code	The response code returned by the web server for the simulated request.	Number	A value between 200 and 300 indicates a good response. A 4xx value indicates a problem with the requested content (eg., page not found). A 5xx value indicates a server error.
Service availability	Indicates whether/not the web service is available.	Percent	A value of 100 indicates that the web service is available and a value of 0 indicates that the web service is not available.
Operation status	Indicates whether/not the configured operation is present in the web service.		This measure will not report metrics if the Operations parameter in the test configuration page is none in the test configuration page.
Operation Content length	Indicates the response code returned by the server for the simulated request.	Number	A value between 200 and 300 indicates a good response. A 4xx value indicates a problem with the requested content (e.g., page not found). A 5xx value indicates a server error. This measure will not report metrics if

Measurement	Description	Measurement Unit	Interpretation
			the Operations parameter in the test configuration page is none or if an invalid Value is specified or if the Value is not specified in the HTML View tab while configuring the operation for monitoring in the test configuration page.
Operation Content validity	This measure validates whether the operation was successful in executing the request made to it.	Percent	A value of 100% indicates that the content returned by the test is valid. A value of 0% indicates that the content may not be valid. This capability for content validation is especially important for multi-tier web applications. For example, a user may not be able to login to the web site but the server may reply back with a valid HTML page where in the error message, say, “Invalid Login” is reported. In this case, the availability will be 100 % (since we got a valid HTML response). If the test is configured such that the content parameter should exclude the string “Invalid Login”, in the above scenario content validity would have a value 0. This measure will not report metrics if the Operations parameter in the test configuration page is none or if an invalid Value is specified or if the Value is not specified in the **HTML View** tab while configuring the operation for monitoring in the test configuration page.
Operation execution time	Indicates the time taken to invoke the configured operation in the web service.	Secs	This measure will not report metrics if the Operations parameter in the test configuration page is none or if an invalid Value is specified or if the Value is not specified in the **HTML View** tab

Measurement	Description	Measurement Unit	Interpretation
			while configuring the operation for monitoring in the test configuration page.

3.1.5.1 Configuring Multiple WSDL URLs for Monitoring

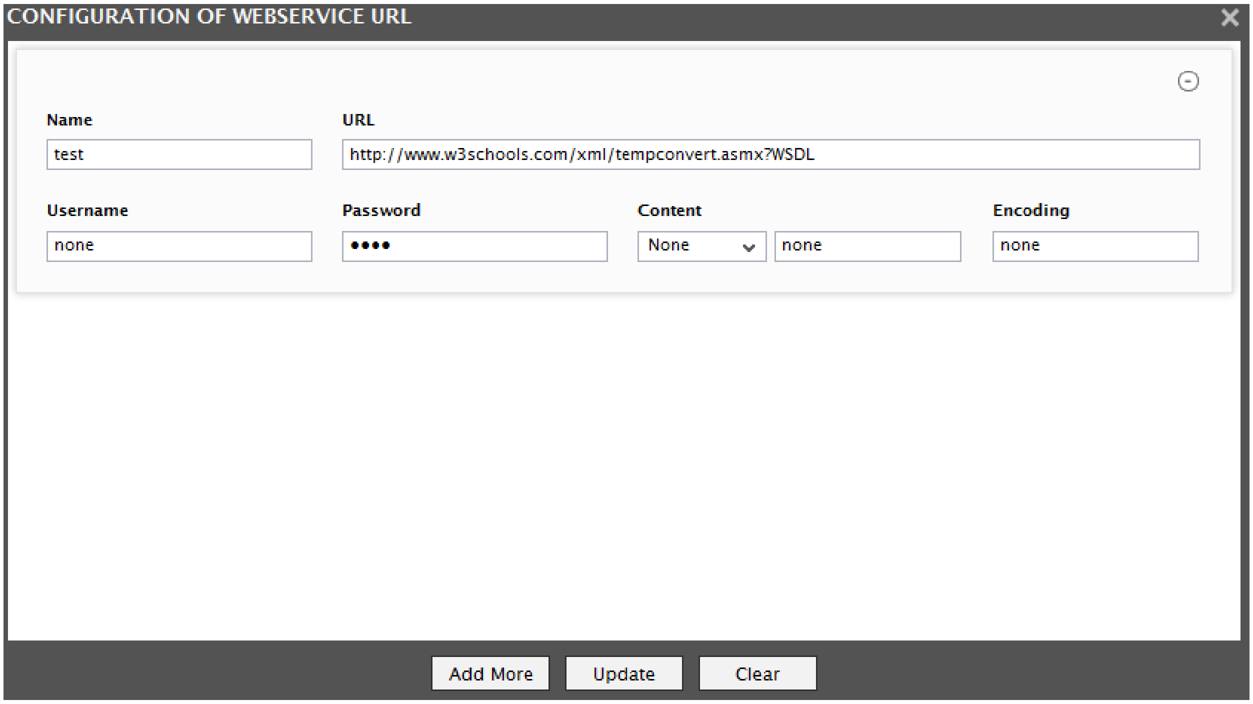
In order to enable the eG agent to connect to multiple WSDL URLs and pull out the required metrics from them, the eG administrative interface provides a special page using which different WSDL URLs and their corresponding operations that need to be monitored can be specified. To configure the WSDL URLs, do the following:


WebService parameters to be configured for jboss:9990 (JBoss AS/EAP)

TEST PERIOD	5 mins
HOST	192.168.10.1
PORT	9990
* WSDL URL	test:http://www.w3schools.com/xml/tempconv
OPERATIONS	test:TempConvert_FahrenheitToCelsius
TIMEOUT	30
DETAILED DIAGNOSIS	● On ○ Off

Figure 3.3: Configuring the WebService test

1. Click on the encircled '+' button alongside the WSDL URL text box. Figure 3.4 will then appear.



CONFIGURATION OF WEBSERVICE URL

Name: test

URL: http://www.w3schools.com/xml/tempconvert.asmx?WSDL

Username: none

Password: ••••

Content: None none

Encoding: none

Figure 3.4: The WebService URL Configuration page

2. Specify the following in Figure 3.4:

- **Name:** Specify a unique name by which the WSDL URL you will be specifiying shortly will be referred to across the eG user interface. This is the name that will appear as the descriptor of this test.
- **URL:** Enter the WSDL URL of the web service that this test should access.
- **Username** and **Password:** These parameters are to be set only if a specific user name / password has to be specified to login to the web service (i.e., WSDL URL) that you have configured for monitoring. In this case, provide valid login credentials using the **Username** and **Password** text boxes. If the server on which **WebService** test executes supports 'Anonymous user access', then these parameters will take either of the following values:
- A valid **Username** and **Password** for the configured WSDL URL
- *none* in both the **Username** and **Password** text boxes of the configured WSDL URL, if no user authorization is required
- Some servers however, support NTLM (Integrated Windows) authentication, where valid login credentials are mandatory. In other words, a *none* specification will not be supported by such

servers. Therefore, in this case, against each configured WSDL URL, you will have to provide a valid **Username** in the format: *domainname\username*, followed by a valid **Password**.

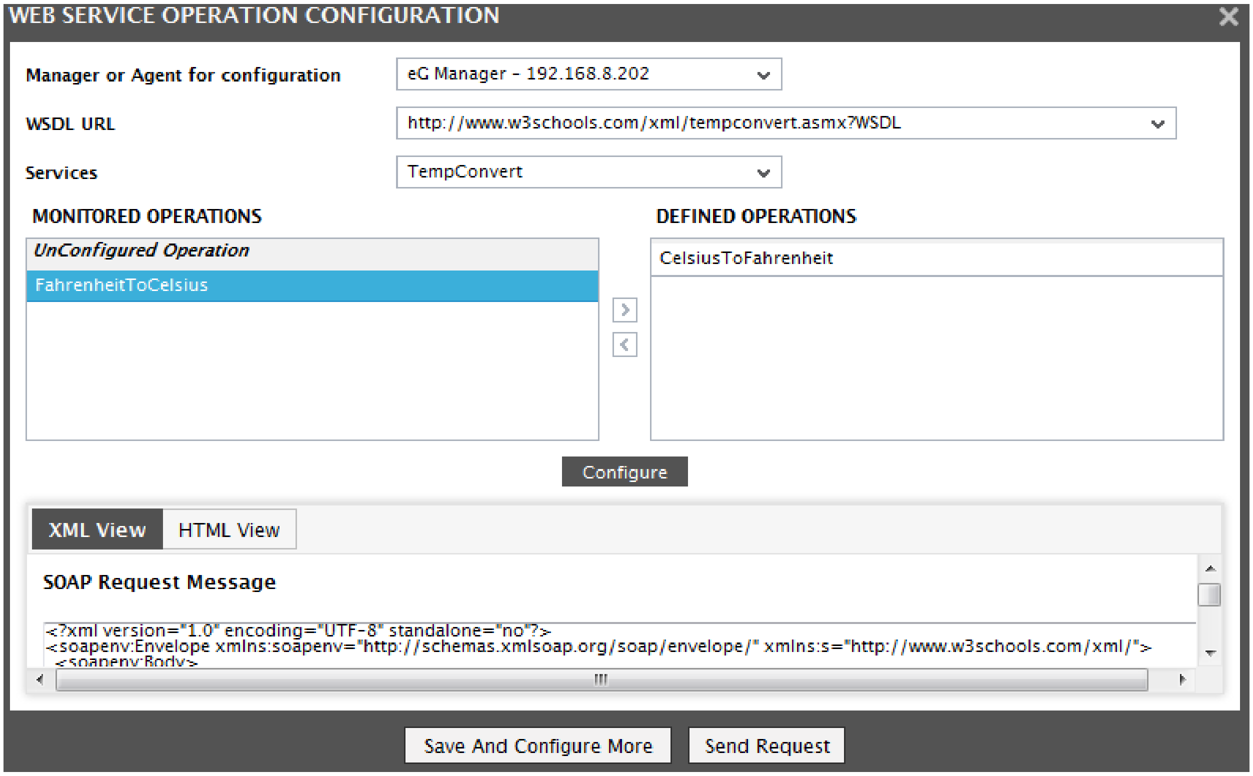
- Please be sure to check if your web service requires HTTP authentication while configuring this parameter. HTTP authentication typically involves a separate pop-up window when you try to access the page. Many services use HTTP POST for obtaining the user name and password and validating the user login. In such cases, the username and password have to be provided as part of the POST information and NOT as part of the **CREDENTIALS** specification for the Web Service test.
- **Content**: The **Content** parameter has to be configured with an instruction:value pair that will be used to validate the content being returned by the test. If the **Content** value is *None*, no validation is performed. On the other hand, if you pick the *Include* option from the **Content** list, it indicates to the test that for the content returned by the web server to be valid, the content must include the specified value (a simple string search is done in this case). This value should be specified in the adjacent text box. Similarly, if the *Exclude* option is chosen from the **Content** drop-down, it indicates to the test that the server's output is valid if it does not contain the value specified in the adjacent text box. The *Include* or *Exclude* value you specify in the text box can include wildcard characters. For example, an *Include* instruction can be **Home page**.
- **Encoding**: Sometimes the eG agent has to parse the WSDL URL content with specific encoding other than the default (ISO-8859-1) encoding. In such a case, specify the type of encoding using which the eG agent can parse the WSDL URL content in the **Encoding** text box. By default, this value is *none*.

3. Similarly, you can add multiple URL specifications by clicking the **Add More** button. To remove a WSDL URL specification, click on the encircled '-' button corresponding to it. To clear all **WSDL URL** specifications, click the **Clear** button. To update all the changes you made, click the **Update** button.
4. Once **Update** is clicked, you will return to the test configuration page as shown in Figure 3.4. The **WSDL URL** text box in the test configuration page will display just the **Name**s - i.e., the unique display names - that you may have configured for the multiple WSDL URLs, as a comma-separated list. To view the complete WSDL URL specification, click the encircled '+' button alongside the **WSDL URL** text box, once again.

3.1.5.2 Configuring Multiple Operations for Monitoring - WebServiceTest

By default, the **Web Service** test will be configured with the WSDL URLs that offer the web services that are to be monitored. To configure the operations that are offered by the WSDL URLs, do the following:

1. Click on the encircled '+' button alongside the Operations text box as shown in Figure 3.3. Figure 3.5 will then appear.



Manager or Agent for configuration: eG Manager - 192.168.8.202

WSDL URL: http://www.w3schools.com/xml/tempconvert.asmx?WSDL

Services: TempConvert

MONITORED OPERATIONS

UnConfigured Operation

FahrenheitToCelsius

DEFINED OPERATIONS

CelsiusToFahrenheit

SOAP Request Message

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:s="http://www.w3schools.com/xml/">
<soapenv:Body>
```

Figure 3.5: Configuring the Web Service Operation

2. Specify the following in Figure 3.5:

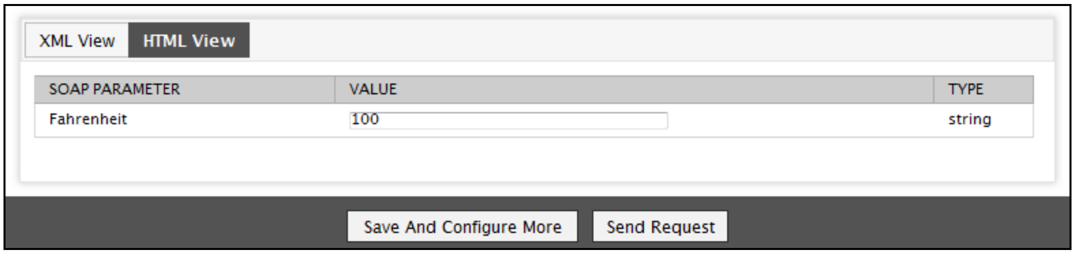
   - Manager/Agent for accessing WSDL URL: Select the eG agent or the eG Manager that is authorized to access the configured WSDL URL from this list.
   - WSDL URL: Once the eG agent/eG Manager is chosen from the Manager/Agent for accessing WSDL URL list, this list will be populated automatically with all the WSDL URLs specified in the WSDL URL text box (See Figure 3.3). Select the WSDL URL of your choice from this list.
   - Services: The web services offered by the chosen WSDL URL will then be populated in this list. Select a service of your choice from this list.

- The operations that are offered by the chosen service will then be populated in the **DEFINED OPERATIONS** list. To monitor a chosen operation, select the operation and click the < button. This will move the chosen operation to the **MONITORED OPERATIONS** list.

- Click the **Configure** button to save the changes.

- The eG agent uses SOAP requests to obtain the necessary metrics from the web service. Once the operation is configured, the XML View of the SOAP Request corresponding to the chosen operation will be generated and listed in the **XML View** tab. Likewise, the **HTML View** tab lists the **SOAP Parameter** that is passed to collect the required metrics for the chosen operation.

- To obtain operation-level statistics, it is important to specify a valid value in the **VALUE** text box of the **HTML View** tab as shown in Figure 3.6. Each time the test is executed, this value will be provided as an input to the chosen operation.

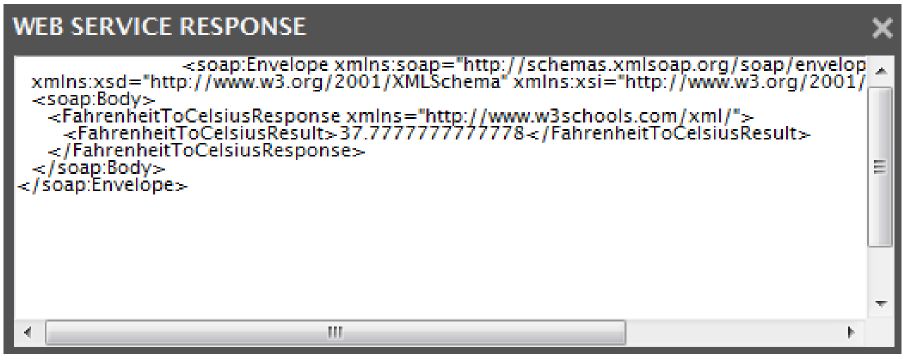


SOAP PARAMETER	VALUE	TYPE
Fahrenheit	100	string

Figure 3.6: Specifying the value for the chosen operation

- Click the **Save and Cofigure More** button to save the changes made.

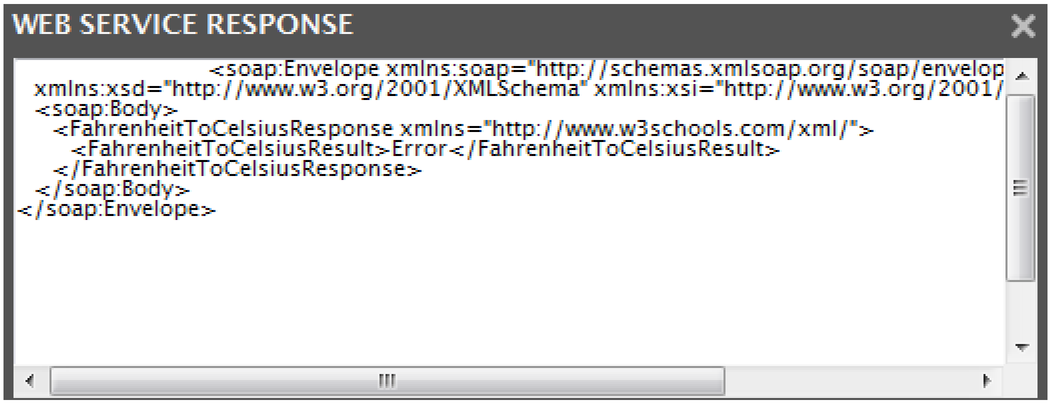
- If you wish to verify if the **VALUE** specified in the **HTML View** tab is valid, then you can do so by clicking the **Send Request** button. Figure 3.7 will then appear. If the value specified in the **VALUE** text box is indeed valid, then the operation will be performed on the value and the result will be specified. For example, if your chosen operation is *FahrenheittoCelsius*, the SOAP Parameter is *Farenheit* and the value that you wish to convert is *100*, the result will be specified in the **WEB SERVICE RESPONSE** pop up window as below: *<FahrenheitToCelsiusResult>37.7777777777778</FahrenheitToCelsiusResult>*



```
                    <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelop
 xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/
 <soap:Body>
   <FahrenheitToCelsiusResponse xmlns="http://www.w3schools.com/xml/">
     <FahrenheitToCelsiusResult>37.7777777777778</FahrenheitToCelsiusResult>
   </FahrenheitToCelsiusResponse>
 </soap:Body>
</soap:Envelope>
```

Figure 3.7: The value that appears when the operation is performed successfully

- If you have specified an invalid value, then a message as follows will be displayed in the pop up window: *<FahrenheitToCelsiusResult>Error</FahrenheitToCelsiusResult>*



```
                    <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelop
 xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/
 <soap:Body>
   <FahrenheitToCelsiusResponse xmlns="http://www.w3schools.com/xml/">
     <FahrenheitToCelsiusResult>Error</FahrenheitToCelsiusResult>
   </FahrenheitToCelsiusResponse>
 </soap:Body>
</soap:Envelope>
```

Figure 3.8: An Error appearing during value conversion

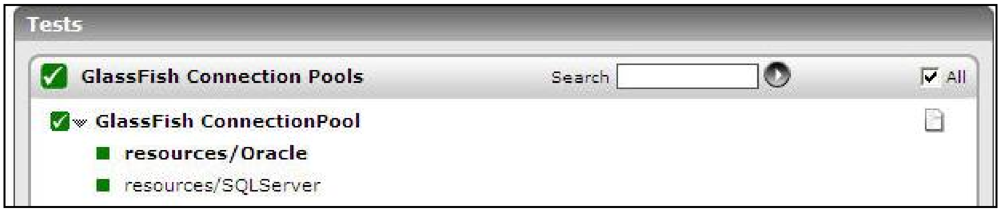
- If you do not specify a **VALUE** or specify an invalid value, operation-level statistics will not be collected by the eG agent and such metrics will not be available in the eG monitoring interface.

3. Similarly, you can configure multiple Operations by clicking the **Configure** button in Figure 3.5. To remove an operation, select the operation from the **MONITORED OPERATION** list and click the **>** button.

Once **Save and Configure More** button is clicked, you will return to the test configuration page (see Figure 3.4). The Operations text box in the test configuration page will display just the operations, as a comma-separated list. To view the complete operation specification, click the encircled '+' button alongside the Operations text box, once again.

3.2 The GlassFish Connection Pools Layer

A connection pool is a cache of database connections maintained so that the connections can be reused when future requests to the database are required. With the help of the **GlassFish ConnectionPool** test mapped to it, this layer reveals how each connection pool configured on the server utilizes the connections in it.


Tests

☑ GlassFish Connection Pools Search ☑ All

☑ GlassFish ConnectionPool

🟩 resources/Oracle

🟩 resources/SQLServer

Figure 3.9: The tests mapped to the GlassFish Connection Pools layer

3.2.1 GlassFish ConnectionPool Test

A connection pool is a cache of database connections maintained so that the connections can be reused when future requests to the database are required. Connection pools are used to enhance the performance of executing commands on a database. Opening and maintaining a database connection for each user, especially requests made to a dynamic database- driven website application, is costly and wastes resources. In connection pooling, after a connection is created, it is placed in the pool and it is used over again so that a new connection does not have to be established. Connection pooling also cuts down on the amount of time a user must wait to establish a connection to the database. As opposed to this, the lack of adequate connections in the pool can prolong the time taken to connect to a database. To analyze the impact of connection pools on the database connection time, use the **GlassFish ConnectionPool** test. This test auto-discovers the connection pools configured on the GlassFish server, tracks the usage of connections in each pool, and reports whether/not sufficient connections are available in the pools to handle the connection requests to the GlassFish server. This way, the test points you to connection pools that are starved for connections and helps you assess how this impacts connection requests to the GlassFish server.

Target of the test : A GlassFish server

Agent deploying the test : An internal/remote agent

Outputs of the test : One set of results for each connection pool on the Java application being monitored.

Configurable parameters for the test

Parameters	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
JMX Remote Port	To collect metrics from a GlassFish Enterprise server, the eG agent monitoring that server should be configured to use **JMX** to connect to the JRE used by the server and pull out the metrics of interest. By default, **JMX** support is enabled for the JRE used by the GlassFish Enterprise Server. The JMX connector listens on port 8686 by default. Therefore, type 8686 as the JMX Remote Port. If JMX listens on a different port in your environment, then specify the same here. To know the port at which JMX listens, open the **domain.xml** file in the <GLASSFISH_INSTALL_DIR>\Sun\AppServer\domains\domain1\config directory and look for the code block shown below: `<jmx-connector accept-all="false" address="0.0.0.0" auth-realm-name="admin-realm" enabled="true" name="system" port="8686" protocol="rmi_jrmp" security-enabled="false">` The port number specified against the Port parameter in the code block above refers to the JMX Remote Port.
JMX User, JMX Password, and Confirm Password	By default, the JMX connector on the GlassFish Enterprise Server requires authentication. To enable the eG agent to use JMX therefore, you need to configure the agent with the credentials of a user who is authorized to use JMX. Typically, the GlassFish administrator has the right to use JMX. You can hence configure the JMX User and JMX Password parameters with the credentials of the *administrator*. However, if you prefer not to expose the credentials of an administrator owing to security considerations, you can use the credentials of any other user with access rights to JMX. To know the name of such a user, open the **domain.xml** file in the <GLASSFISH_INSTALL_DIR>\Sun\AppServer\domains\domain1\config directory and look for the code block shown below: `<jmx-connector accept-all="false" address="0.0.0.0" auth-realm-name="admin-realm" enabled="true" name="system" port="8686" protocol="rmi_jrmp" security-enabled="false">` The user name specified against the auth-realm-name parameter in the code block above can be configured as the JMX User, and the Password of that user can be specified against JMX Password. Confirm the JMX Password you specify by retyping that password in the Confirm Password text box.
JNDIName	The JNDIName is a lookup name for connecting to the JMX connector. By default, this

Parameters	Description
	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 Provider	This test uses a JMX Provider to access the MBean attributes of the GlassFish Enterprise 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 GlassFish Enterprise 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.

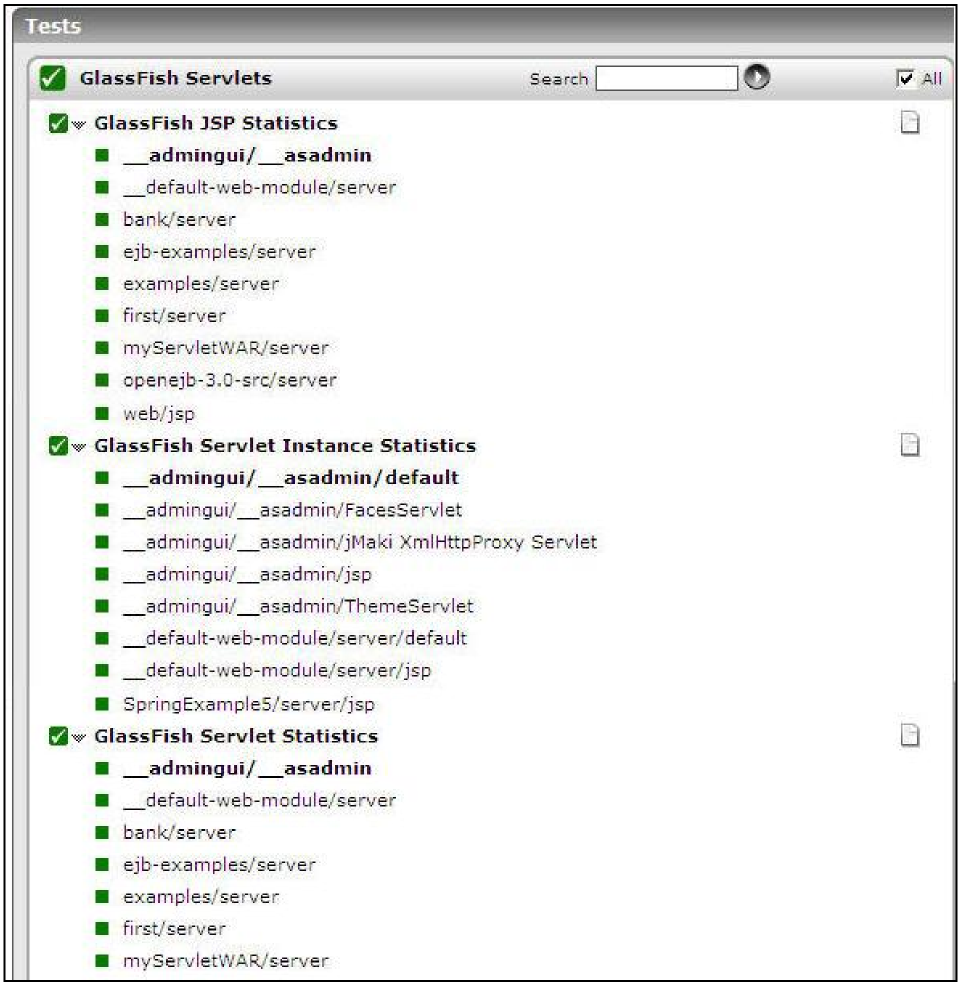
Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Total beans in cache	Indicates the number of EJBs in the EJB cache.	Number	
Used connections	Indicates currently used connections in this pool.	Number	A high value indicates a large number of connections to the database.
Free connections	Indicates the total number of free connections in this pool.	Number	
Connection waiting time	Indicates waiting time of the last request that was serviced in the pool.	Secs	
Average connection waiting time	Indicates average waiting time per successful request to this pool.	Secs	The effectiveness of a connection pool can be measured by observing how much it reduces the connection time. A low value for this measure is hence a clear indicator that the connection pool is effective in reducing the wait time of connection requests. A high value on the other hand, could indicate that the pool is running out of connections often, causing connection requests to wait for long periods of time to be processed.
Connections timedout	Indicates the total number of connections in this pool	Number	

Measurement	Description	Measurement Unit	Interpretation
	that timed out between the start time and the last sample time.		
Potential connection leak	Indicates currently available potential connection leaks.	Number	Ideally, the value of this measure should be 0.
Waiting queue length	Indicates the number of connection requests to this pool that are currently in waiting queue.	Number	A consistent increase in the value of this measure could indicate that there are not enough connections in the pool to service the connection requests received by the GlassFish server.
Failed connections	Indicates number of connections in this connection pool that failed validation from the start time until the last sample time.	Number	Ideally, the value of this measure should be 0.
Released connections	Indicates the current number of logical connections released to this pool.	Number	
Destroyed connections	Indicates the number of physical connections in this pool that were destroyed since the last reset.	Number	

3.3 The GlassFish Servlets Layer

The tests mapped to this layer monitors the ability of the GlassFish server to process requests for JSPs and servlets. In the event of a slowdown of one/more web applications deployed on the server, you can use these tests to figure out where the slowdown occurred - while processing requests for JSPs? or while processing servlets? - and also isolate the JSP/servlet that is delaying the processing.



☑ GlassFish Servlets

- ☑ GlassFish JSP Statistics
  - 🟩 **__admingui/__asadmin**
  - 🟩 __default-web-module/server
  - 🟩 bank/server
  - 🟩 ejb-examples/server
  - 🟩 examples/server
  - 🟩 first/server
  - 🟩 myServletWAR/server
  - 🟩 openejb-3.0-src/server
  - 🟩 web/jsp
- ☑ GlassFish Servlet Instance Statistics
  - 🟩 **__admingui/__asadmin/default**
  - 🟩 __admingui/__asadmin/FacesServlet
  - 🟩 __admingui/__asadmin/jMaki XmlHttpProxy Servlet
  - 🟩 __admingui/__asadmin/jsp
  - 🟩 __admingui/__asadmin/ThemeServlet
  - 🟩 __default-web-module/server/default
  - 🟩 __default-web-module/server/jsp
  - 🟩 SpringExample5/server/jsp
- ☑ GlassFish Servlet Statistics
  - 🟩 **__admingui/__asadmin**
  - 🟩 __default-web-module/server
  - 🟩 bank/server
  - 🟩 ejb-examples/server
  - 🟩 examples/server
  - 🟩 first/server
  - 🟩 myServletWAR/server

Figure 3.10: The tests mapped to the GlassFish Servlets layer

3.3.1 GlassFish JSP Statistics Test

To know how well the GlassFish server processes requests for JSP pages and to promptly capture JSP errors, use this test.

Target of the test : A GlassFish server

Agent deploying the test : An internal/remote agent

Outputs of the test : One set of results for each JSP processed by the GlassFish Enterprise server being monitored.

Configurable parameters for the test

Parameters	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
JMX Remote Port	To collect metrics from a GlassFish Enterprise server, the eG agent monitoring that server should be configured to use **JMX** to connect to the JRE used by the server and pull out the metrics of interest. By default, **JMX** support is enabled for the JRE used by the GlassFish Enterprise Server. The JMX connector listens on port 8686 by default. Therefore, type 8686 as the JMX Remote Port. If JMX listens on a different port in your environment, then specify the same here. To know the port at which JMX listens, open the **domain.xml** file in the <GLASSFISH_INSTALL_DIR>\Sun\AppServer\domains\domain1\config directory and look for the code block shown below: `<jmx-connector accept-all="false" address="0.0.0.0" auth-realm-name="admin-realm" enabled="true" name="system" port="8686" protocol="rmi_jrmp" security-enabled="false">` The port number specified against the Port parameter in the code block above refers to the JMX Remote Port.
JMX User, JMX Password, and Confirm Password	By default, the JMX connector on the GlassFish Enterprise Server requires authentication. To enable the eG agent to use JMX therefore, you need to configure the agent with the credentials of a user who is authorized to use JMX. Typically, the GlassFish administrator has the right to use JMX. You can hence configure the JMX User and JMX Password parameters with the credentials of the *administrator*. However, if you prefer not to expose the credentials of an administrator owing to security considerations, you can use the credentials of any other user with access rights to JMX. To know the name of such a user, open the **domain.xml** file in the <GLASSFISH_INSTALL_DIR>\Sun\AppServer\domains\domain1\config directory and look for the code block shown below: `<jmx-connector accept-all="false" address="0.0.0.0" auth-realm-name="admin-realm" enabled="true" name="system" port="8686" protocol="rmi_jrmp" security-enabled="false">` The user name specified against the auth-realm-name parameter in the code block above can be configured as the JMX User, and the Password of that user can be specified against JMX Password. Confirm the JMX Password you specify by retyping that password in the Confirm Password text box.
JNDIName	The JNDIName is a lookup name for connecting to the JMX connector. By default, this

Parameters	Description
	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 Provider	This test uses a JMX Provider to access the MBean attributes of the GlassFish Enterprise 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 GlassFish Enterprise 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
Active jsps	Indicates the number of JSPs that are currently active on this application.	Number	
Jsp loaded	Indicates the number of JSPs that were loaded by this application during the last measurement period.	Number	This counter gets incremented the first time a JSP is accessed, and its corresponding servlet is loaded and initialized.
Jsp reloaded	Indicates the number of JSPs that were reloaded by this application during the last measurement period.	Number	This counter gets incremented whenever a JSP whose source code has changed since it was first deployed is accessed again and recompiled, and its corresponding servlet is reloaded and reinitialized.
Jsp errors	Indicates the number of JSP errors experienced by this application during the last measurement period.	Number	Ideally, the value of this measure should be 0.

3.3.2 GlassFish Servlet Instance Statistics Test

Whenever users to a web application deployed on the GlassFish Enterprise server complain of a slowdown, administrators need to instantly determine what is causing the slowdown. The **GlassFish Servlet Statistics** test will enable administrators to figure out if the servlets used by the problem application is contributing to the slowdown or not. If so, then, administrators can use the

GlassFish Servlet Instance Statistics test to isolate that servlet, which is the processing bottleneck. This test auto-discovers the servlets used by each web application that is installed on the GlassFish server, and reports the processing time of and errors experienced by each servlet, so that problematic servlets can be quickly and accurately isolated, and the root-cause of the problems investigated.

By default, this test monitors only those servlets that are part of a servlet group. To configure servlet groups, click on the 'Click here' hyperlink above the parameters of this test.

Target of the test : A GlassFish server

Agent deploying the test : An internal/remote agent

Outputs of the test : One set of results for each servlet group/servlet used by each web application deployed on the GlassFish Enterprise server being monitored.

Configurable parameters for the test

Parameters	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
JMX Remote Port	To collect metrics from a GlassFish Enterprise server, the eG agent monitoring that server should be configured to use **JMX** to connect to the JRE used by the server and pull out the metrics of interest. By default, **JMX** support is enabled for the JRE used by the GlassFish Enterprise Server. The JMX connector listens on port 8686 by default. Therefore, type 8686 as the JMX Remote Port. If JMX listens on a different port in your environment, then specify the same here. To know the port at which JMX listens, open the **domain.xml** file in the <GLASSFISH_INSTALL_DIR>\Sun\AppServer\domains\domain1\config directory and look for the code block shown below: `<jmx-connector accept-all="false" address="0.0.0.0" auth-realm-name="admin-realm" enabled="true" name="system" port="8686" protocol="rmi_jrmp" security-enabled="false">` The port number specified against the Port parameter in the code block above refers to the JMX Remote Port.
JMX User, JMX Password, and Confirm Password	By default, the JMX connector on the GlassFish Enterprise Server requires authentication. To enable the eG agent to use JMX therefore, you need to configure the agent with the credentials of a user who is authorized to use JMX. Typically, the

Parameters	Description
	GlassFish administrator has the right to use JMX. You can hence configure the JMX User and JMX Password parameters with the credentials of the *administrator*. However, if you prefer not to expose the credentials of an administrator owing to security considerations, you can use the credentials of any other user with access rights to JMX. To know the name of such a user, open the **domain.xml** file in the <GLASSFISH_INSTALL_DIR>\Sun\AppServer\domains\domain1\config directory and look for the code block shown below: `<jmx-connector accept-all="false" address="0.0.0.0" auth-realm-name="admin-realm" enabled="true" name="system" port="8686" protocol="rmi_jrmp" security-enabled="false">` The user name specified against the auth-realm-name parameter in the code block above can be configured as the JMX User, and the Password of that user can be specified against JMX Password. Confirm the JMX Password you specify by retyping that password in the Confirm Password text box.
JNDIName	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 Provider	This test uses a JMX Provider to access the MBean attributes of the GlassFish Enterprise 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 GlassFish Enterprise 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
Requests processed	Indicates the number of requests to this application that were processed by this servlet group/servlet during the last measurement period.	Number	This measure is a good indicator of the load on each servlet. Compare the value of this measure across the servlets of an application to identify which servlet is experiencing very high load.
Avg processing time	Indicates the time taken by this servlet group/servlet to process requests.	Secs	A low value is desired for this measure. A consistent rise in the value of this measure could indicate a

Measurement	Description	Measurement Unit	Interpretation
			processing bottleneck, which in turn may affect application performance. Compare the value of this measure across servlets (associated with a particular application) to know which servlet processing requests too slowly.
Maximum time	Indicates the maximum time taken by this servlet group/servlet to process requests.	Secs	
Servlet errors	Indicates the number of error responses that were sent by this servlet group/servlet during the last measurement period.	Number	This count includes the following: • The number of requests for which responses with response code that is greater than or equal to 400 were sent; • The number of requests for which a response could not be sent at all. Ideally, the value of this measure should be 0.
Error	Indicates the percentage of error responses that were sent by this servlet group/servlet during the last measurement period	Percent	Ideally, the value of this measure should be 0.

3.3.3 GlassFish Servlet Statistics Test

For each web application that is deployed on a GlassFish server, this test monitors the processing ability of the servlets used by that application, and reports whether the servlets are responsible for any application slowdowns noticed by users.

Target of the test : A GlassFish server

Agent deploying the test : An internal/remote agent

Outputs of the test : One set of results for each web application deployed on the GlassFish Enterprise server being monitored.

Configurable parameters for the test

Parameters	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
JMX Remote Port	To collect metrics from a GlassFish Enterprise server, the eG agent monitoring that server should be configured to use **JMX** to connect to the JRE used by the server and pull out the metrics of interest. By default, **JMX** support is enabled for the JRE used by the GlassFish Enterprise Server. The JMX connector listens on port 8686 by default. Therefore, type 8686 as the JMX Remote Port. If JMX listens on a different port in your environment, then specify the same here. To know the port at which JMX listens, open the **domain.xml** file in the <GLASSFISH_INSTALL_ DIR>\Sun\AppServer\domains\domain1\config directory and look for the code block shown below: `<jmx-connector accept-all="false" address="0.0.0.0" auth-realm-name="admin-realm" enabled="true" name="system" port="8686" protocol="rmi_jrmp" security-enabled="false">` The port number specified against the Port parameter in the code block above refers to the JMX Remote Port.
JMX User, JMX Password, and Confirm Password	By default, the JMX connector on the GlassFish Enterprise Server requires authentication. To enable the eG agent to use JMX therefore, you need to configure the agent with the credentials of a user who is authorized to use JMX. Typically, the GlassFish administrator has the right to use JMX. You can hence configure the JMX User and JMX Password parameters with the credentials of the *administrator*. However, if you prefer not to expose the credentials of an administrator owing to security considerations, you can use the credentials of any other user with access rights to JMX. To know the name of such a user, open the **domain.xml** file in the <GLASSFISH_INSTALL_DIR>\Sun\AppServer\domains\domain1\config directory and look for the code block shown below: `<jmx-connector accept-all="false" address="0.0.0.0" auth-realm-name="admin-realm" enabled="true" name="system" port="8686" protocol="rmi_jrmp" security-enabled="false">` The user name specified against the auth-realm-name parameter in the code block above can be configured as the JMX User, and the Password of that user can be

Parameters	Description
	specified against JMX Password. Confirm the JMX Password you specify by retyping that password in the Confirm Password text box.
JNDIName	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 Provider	This test uses a JMX Provider to access the MBean attributes of the GlassFish Enterprise 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 GlassFish Enterprise 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

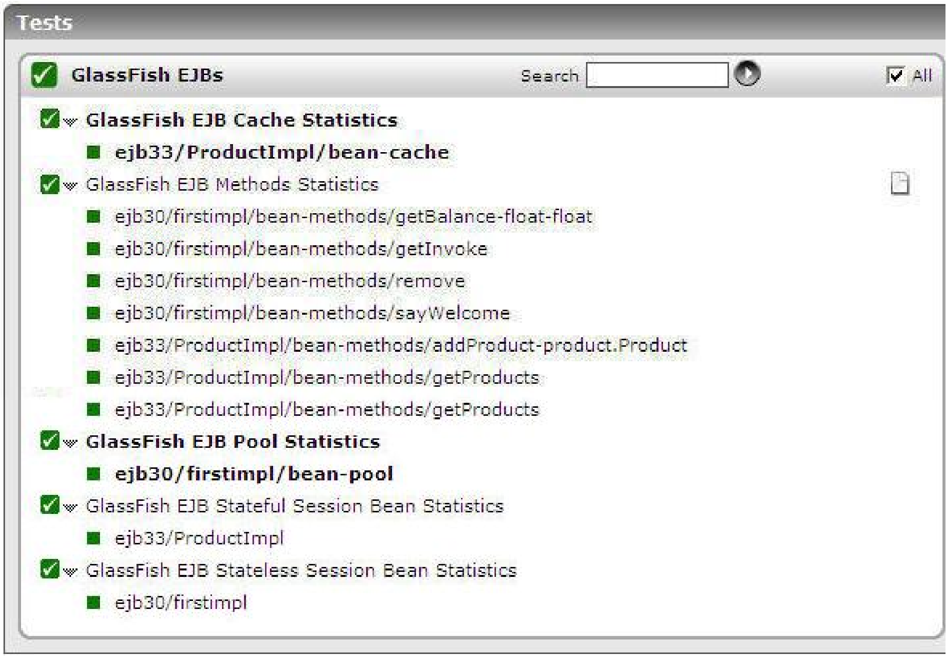
Measurement	Description	Measurement Unit	Interpretation
Active servlets	Indicates the number of servlets of this application that are actively servicing requests currently.	Number	
Servlet processing time	Indicates the time taken by the servlets of this application to process requests during the last measurement period.	Secs	A low value is desired for this measure. A consistent rise in the value of this measure could indicate that the servlets are taking too long to process requests.
Servlets loaded	Indicates the number of servlets loaded into this application during the last measurement period.	Number	

3.4 The GlassFish EJBs Layer

The tests mapped to this layer enable monitoring of the following:

- Usage of EJB caches;
- EJB methods and their execution times;

- Usage of beans in the EJB pools;
- The Stateless and Stateful session bean containers and the nature of methods invoked on them.



Tests

☑ GlassFish EJBs Search ☑ All

- ☑ GlassFish EJB Cache Statistics
  - 🟩 ejb33/ProductImpl/bean-cache
- ☑ GlassFish EJB Methods Statistics
  - 🟩 ejb30/firstimpl/bean-methods/getBalance-float-float
  - 🟩 ejb30/firstimpl/bean-methods/getInvoke
  - 🟩 ejb30/firstimpl/bean-methods/remove
  - 🟩 ejb30/firstimpl/bean-methods/sayWelcome
  - 🟩 ejb33/ProductImpl/bean-methods/addProduct-product.Product
  - 🟩 ejb33/ProductImpl/bean-methods/getProducts
  - 🟩 ejb33/ProductImpl/bean-methods/getProducts
- ☑ GlassFish EJB Pool Statistics
  - 🟩 ejb30/firstimpl/bean-pool
- ☑ GlassFish EJB Stateful Session Bean Statistics
  - 🟩 ejb33/ProductImpl
- ☑ GlassFish EJB Stateless Session Bean Statistics
  - 🟩 ejb30/firstimpl

Figure 3.11: The tests mapped to the GlassFish EJBs layer

3.4.1 GlassFish EJB Cache Statistics

To improve the performance and the scalability of the EJB container, the GlassFish server caches EJBs. A bean in the cache represents the ready state in the EJB lifecycle. This means that the bean has an identity (for example, a primary key or session ID) associated with it.

Beans moving out of the cache have to be passivated or destroyed according to the EJB lifecycle. Once passivated, a bean has to be activated to come back into the cache.

Any incoming request using these "ready" beans from the cache avoids the overhead of creation, setting identity, and potentially activation. It is hence good to cache as many beans as possible.

Using this test, you can monitor the usage of the EJB cache and accurately identify caches that are grossly under-utilized. You can also diagnose the probable cause for the ineffective cache utilization - is it because adequate beans are not available in the cache? if so, why? - is it because the cache size is too small and needs to be resized to accommodate more beans or to reduce cache overflow? In the process, you can even determine the success/failure of stateful session bean passivation attempts.

Target of the test : A GlassFish server

Agent deploying the test : An internal/remote agent

Outputs of the test : One set of results for each cache configured on the GlassFish Enterprise server being monitored.

Configurable parameters for the test

Parameters	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
JMX Remote Port	To collect metrics from a GlassFish Enterprise server, the eG agent monitoring that server should be configured to use **JMX** to connect to the JRE used by the server and pull out the metrics of interest. By default, **JMX** support is enabled for the JRE used by the GlassFish Enterprise Server. The JMX connector listens on port 8686 by default. Therefore, type 8686 as the JMX Remote Port. If JMX listens on a different port in your environment, then specify the same here. To know the port at which JMX listens, open the **domain.xml** file in the <GLASSFISH_INSTALL_ DIR>\Sun\AppServer\domains\domain1\config directory and look for the code block shown below: `<jmx-connector accept-all="false" address="0.0.0.0" auth-realm-name="admin-realm" enabled="true" name="system" port="8686" protocol="rmi_jrmp" security-enabled="false">` The port number specified against the Port parameter in the code block above refers to the JMX Remote Port.
JMX User, JMX Password, and Confirm Password	By default, the JMX connector on the GlassFish Enterprise Server requires authentication. To enable the eG agent to use JMX therefore, you need to configure the agent with the credentials of a user who is authorized to use JMX. Typically, the GlassFish administrator has the right to use JMX. You can hence configure the JMX User and JMX Password parameters with the credentials of the *administrator*. However, if you prefer not to expose the credentials of an administrator owing to security considerations, you can use the credentials of any other user with access rights to JMX. To know the name of such a user, open the **domain.xml** file in the <GLASSFISH_INSTALL_DIR>\Sun\AppServer\domains\domain1\config directory and look for the code block shown below: `<jmx-connector accept-all="false" address="0.0.0.0" auth-realm-name="admin-realm" enabled="true" name="system" port="8686" protocol="rmi_jrmp" security-enabled="false">`

Parameters	Description
	The user name specified against the auth-realm-name parameter in the code block above can be configured as the JMX User, and the Password of that user can be specified against JMX Password. Confirm the JMX Password you specify by retyping that password in the Confirm Password text box.
JNDIName	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 Provider	This test uses a JMX Provider to access the MBean attributes of the GlassFish Enterprise 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 GlassFish Enterprise 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
Total beans in cache	Indicates the number of EJBs in this EJB cache.	Number	
Cache hits	Indicates current number of hits in this EJB Cache.	Number	A high value is desired for this measure. A low value indicates that adequate beans are not available in the cache to service requests to the server. This in turn could increase the processing overheads of the requests to the server.
Cache misses	Indicates the number of times a current user request failed to find an EJB in this EJB cache instance.	Number	A low value is desired for this measure.
Cache hits ratio	Indicates the percentage of user requests that were serviced by the beans in this cache.	Percent	A high value is desired for this measure. A low value indicates that adequate beans are not available in the cache to service requests to the server. This in turn could increase the processing overheads of the

Measurement	Description	Measurement Unit	Interpretation
			requests to the server. Under such circumstances, you may want to alter the size of the cache by modifying the max-cache-size and resize-quantity parameters. The max-cache-size governs the maximum number of beans in the cache. Increasing this size can increase the capacity of the cache, and thus enable it to hold additional beans. In case of entity beans, you need to tune this parameter depending on the usage of a particular entity bean. While beans that are used less (for example, an order that is created and never used after the transaction is over) are to be cached less, the beans that are used frequently (for example, an item in the inventory that gets referenced very often), are to be cached more in numbers. The resize-quantity governs the number of beans to be created or deleted when the cache is serviced by the server. By increasing this quantity, you can ensure that additional beans are created to serve additional user requests.
StatefulSessionBean passivations	Indicates the number of StatefulSessionBean passivations that are currently occurring in this cache.	Number	Beans moving out of the cache have to be passivated or destroyed according to the EJB lifecycle. Once passivated, a bean has to be activated to come back into the cache. Typically, bean passivation occurs when the max-cache-size setting is exceeded and the cache overflows. A bean is also passivated if it outlives the cache-

Measurement	Description	Measurement Unit	Interpretation
			idle-timeout-in-seconds settings.
Passivations success	Indicates the number of passivations that suceeded currently in this cache.	Number	
Passivations errors	Indicates the number of errors that occurred when passivating the beans in this cache.	Number	Ideally, the value of this measure should be 0.
Removed sessions	Indicates the number of expired session beans that were currently removed from this cache.	Number	If a stateful session bean remains idle in the backup store for a duration beyond the configured removal-timeout-in-seconds, then it is removed from the backup store and will not be accessible to the client. The default value is 60 minutes.

3.4.2 GlassFish EJB Method Statistics

This test auto-discovers the methods invoked by the GlassFish server, reports the time taken by the server to execute each method, and reveals errors that may have occurred during method execution. When application users complaint of a slowdown, then this test provides the method-level insight that is necessary for troubleshooting the slowdown. With the help of this test, administrators can quickly figure out which method is taking too long to execute and which methods are throwing exceptions often, and will thus be able to zero-in on the root-cause of the application slowdown.

Target of the test : A GlassFish server

Agent deploying the test : An internal/remote agent

Outputs of the test : One set of results for each method invoked by the GlassFish Enterprise server being monitored.

Configurable parameters for the test

Parameters	Description
Test Period	How often should the test be executed.

Parameters	Description
Host	The host for which the test is to be configured.
Port	The port number at which the specified host listens
JMX Remote Port	To collect metrics from a GlassFish Enterprise server, the eG agent monitoring that server should be configured to use **JMX** to connect to the JRE used by the server and pull out the metrics of interest. By default, **JMX** support is enabled for the JRE used by the GlassFish Enterprise Server. The JMX connector listens on port 8686 by default. Therefore, type 8686 as the JMX Remote Port. If JMX listens on a different port in your environment, then specify the same here. To know the port at which JMX listens, open the **domain.xml** file in the <GLASSFISH_INSTALL_ DIR>\Sun\AppServer\domains\domain1\config directory and look for the code block shown below: `<jmx-connector accept-all="false" address="0.0.0.0" auth-realm-name="admin-realm" enabled="true" name="system" port="8686" protocol="rmi_jrmp" security-enabled="false">` The port number specified against the Port parameter in the code block above refers to the JMX Remote Port.
JMX User, JMX Password, and Confirm Password	By default, the JMX connector on the GlassFish Enterprise Server requires authentication. To enable the eG agent to use JMX therefore, you need to configure the agent with the credentials of a user who is authorized to use JMX. Typically, the GlassFish administrator has the right to use JMX. You can hence configure the JMX User and JMX Password parameters with the credentials of the *administrator*. However, if you prefer not to expose the credentials of an administrator owing to security considerations, you can use the credentials of any other user with access rights to JMX. To know the name of such a user, open the **domain.xml** file in the <GLASSFISH_INSTALL_DIR>\Sun\AppServer\domains\domain1\config directory and look for the code block shown below: `<jmx-connector accept-all="false" address="0.0.0.0" auth-realm-name="admin-realm" enabled="true" name="system" port="8686" protocol="rmi_jrmp" security-enabled="false">` The user name specified against the auth-realm-name parameter in the code block above can be configured as the JMX User, and the Password of that user can be specified against JMX Password. Confirm the JMX Password you specify by retyping that password in the Confirm Password text box.
JNDIName	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.

Parameters	Description
JMX Provider	This test uses a JMX Provider to access the MBean attributes of the GlassFish Enterprise 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 GlassFish Enterprise 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
Success methods	Indicates the number of invocations of this method that were successful during the last measurement period.	Number	
Error methods	Indicates the number of invocations of this method that resulted in errors during the last measurement period.	Number	Ideally, the value of this measure should be 0.
Error	Indicates the percentage of this method's invocations that resulted in errors.	Percent	A very low value is desired for this measure. Too many errors in method execution can adversely impact application performance.
Execution time	Indicates the time taken for this method to execute during the last measurement period.	Secs	A very low value is desired for this measure. A high value is indicative of problems in method execution, which can cause slowdowns in application performance.
Statistics	Indicates the number of times this method was invoked in the last measurement period.	Number	

3.4.3 GlassFish EJB Pool Statistics Test

A bean in the pool represents the pooled state in the EJB lifecycle. This means that the bean does not have an identity. The advantage of having beans in the pool is that the time to create a bean can be saved for a request. The container has mechanisms that create pool objects in the background, to save the time of bean creation on the request path.

Stateless session beans and entity beans use the EJB pool. By continuously monitoring the usage of these beans in the pools, you can accurately isolate the pools that are over-utilized - i.e., pools that do not enough free beans to service future requests - and under-utilized - i.e., pools that have too many beans than required. This test enables administrators to achieve the same. Based on the usage reports of this test, administrators can tune the pool size, so that abnormal usage patterns can be avoided.

Target of the test : A GlassFish server

Agent deploying the test : An internal/remote agent

Outputs of the test : One set of results for each EJB pool configured on the GlassFish Enterprise server being monitored.

Configurable parameters for the test

Parameters	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
JMX Remote Port	To collect metrics from a GlassFish Enterprise server, the eG agent monitoring that server should be configured to use **JMX** to connect to the JRE used by the server and pull out the metrics of interest. By default, **JMX** support is enabled for the JRE used by the GlassFish Enterprise Server. The JMX connector listens on port 8686 by default. Therefore, type 8686 as the JMX Remote Port. If JMX listens on a different port in your environment, then specify the same here. To know the port at which JMX listens, open the **domain.xml** file in the <GLASSFISH_INSTALL_DIR>\Sun\AppServer\domains\domain1\config directory and look for the code block shown below: `<jmx-connector accept-all="false" address="0.0.0.0" auth-realm-name="admin-realm" enabled="true" name="system" port="8686" protocol="rmi_jrmp" security-enabled="false">`

Parameters	Description
	The port number specified against the Port parameter in the code block above refers to the JMX Remote Port.
JMX User, JMX Password, and Confirm Password	By default, the JMX connector on the GlassFish Enterprise Server requires authentication. To enable the eG agent to use JMX therefore, you need to configure the agent with the credentials of a user who is authorized to use JMX. Typically, the GlassFish administrator has the right to use JMX. You can hence configure the JMX User and JMX Password parameters with the credentials of the *administrator*. However, if you prefer not to expose the credentials of an administrator owing to security considerations, you can use the credentials of any other user with access rights to JMX. To know the name of such a user, open the **domain.xml** file in the <GLASSFISH_INSTALL_DIR>\Sun\AppServer\domains\domain1\config directory and look for the code block shown below: `<jmx-connector accept-all="false" address="0.0.0.0" auth-realm-name="admin-realm" enabled="true" name="system" port="8686" protocol="rmi_jrmp" security-enabled="false">` The user name specified against the auth-realm-name parameter in the code block above can be configured as the JMX User, and the Password of that user can be specified against JMX Password. Confirm the JMX Password you specify by retyping that password in the Confirm Password text box.
JNDIName	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 Provider	This test uses a JMX Provider to access the MBean attributes of the GlassFish Enterprise 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 GlassFish Enterprise 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
Total beans in pool	Indicates the number of beans that are currently available in this pool.	Number	

Measurement	Description	Measurement Unit	Interpretation
Total waiting threads	Indicates the number of threads that are currently waiting for a free bean in this pool.	Number	A well-tuned pool is one that has enough free beans to service all user requests, and does not keep any request thread waiting for a free bean. Ideally therefore, this measure should report the value 0 for a pool. A non-zero value indicates that the corresponding pool is running out of beans. This could be because the pool is improperly sized. You may then have to resize the pool. For this, you can use any of the following parameters: • **steady-pool-size**: This governs the initial and minimum number of beans maintained in the pool. Set this property to a number greater than zero for a moderately loaded system. Having a value greater than zero ensures that there is always a pooled instance to process an incoming request. • **max-pool-size**: This governs the maximum number of connections that can be created to satisfy client requests. Set this property to be representative of the anticipated high load of the system. An very large pool wastes memory and can slow down the system. A very small pool is also inefficient due to contention. • **resize-quantity**: This governs the number of beans to be created or

Measurement	Description	Measurement Unit	Interpretation
			deleted in the pool when the requests are being serviced by the server. Be sure to re-calibrate the pool resize quantity when you change the maximum pool size, to maintain an equilibrium. Generally, a larger maximum pool size should have a larger pool resize quantity.
Loaded JMS messages	Indicates the number of messages that are currently loaded into a JMS session in this pool during the last measurement period.	Number	The message-driven bean container uses the JMS service integrated into the GlassFish Server for message-driven beans that are JMS clients. The container manages a pool of message-driven beans for the concurrent processing of a stream of messages. A message-driven bean is a client to a Connector inbound resource adapter.
Created beans in pool	Indicates the number of beans created in this pool during the last measurement period.	Number	
Destroyed beans in pool	Indicates the number of beans destroyed in this pool during the last measurement period.	Number	If one/more beans in a pool violate the pool-idle-timeout-in-seconds setting of that pool - i.e., if the beans were idle for a duration longer than the pool-idle-timeout-in-seconds configuration - then such beans are destroyed, provided they are stateless session beans or message driver beans.

3.4.4 GlassFish EJB Stateful Session Bean Statistics Test

The stateful container manages the stateful session beans, which, by definition, carry the client-specific state. There is a one-to-one relationship between the client and the stateful session beans. At creation, each stateful session bean (SFSB) is given a unique session ID that is used to access the session bean so that an instance of a stateful session bean is accessed by a single client only. Stateful session beans are managed using cache.

This test monitors the stateful container and reports the number and nature of methods that were invoked on the container. In the process, the test reports the status of the stateful session beans - whether they have just been created, are in the 'ready' state, are in the 'passivate' state, or have been removed from the container.

Target of the test : A GlassFish server

Agent deploying the test : An internal/remote agent

Outputs of the test : One set of results for each stateful session bean container configured on the GlassFish Enterprise server being monitored.

Configurable parameters for the test

Parameters	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
JMX Remote Port	To collect metrics from a GlassFish Enterprise server, the eG agent monitoring that server should be configured to use **JMX** to connect to the JRE used by the server and pull out the metrics of interest. By default, **JMX** support is enabled for the JRE used by the GlassFish Enterprise Server. The JMX connector listens on port 8686 by default. Therefore, type 8686 as the JMX Remote Port. If JMX listens on a different port in your environment, then specify the same here. To know the port at which JMX listens, open the **domain.xml** file in the <GLASSFISH_INSTALL_ DIR>\Sun\AppServer\domains\domain1\config directory and look for the code block shown below: `<jmx-connector accept-all="false" address="0.0.0.0" auth-realm-name="admin-realm" enabled="true" name="system" port="8686" protocol="rmi_jrmp" security-enabled="false">` The port number specified against the Port parameter in the code block above refers to the JMX Remote Port.
JMX User, JMX Password, and Confirm Password	By default, the JMX connector on the GlassFish Enterprise Server requires authentication. To enable the eG agent to use JMX therefore, you need to configure the agent with the credentials of a user who is authorized to use JMX. Typically, the GlassFish administrator has the right to use JMX. You can hence configure the JMX

Parameters	Description
	User and JMX Password parameters with the credentials of the *administrator*. However, if you prefer not to expose the credentials of an administrator owing to security considerations, you can use the credentials of any other user with access rights to JMX. To know the name of such a user, open the **domain.xml** file in the <GLASSFISH_INSTALL_DIR>\Sun\AppServer\domains\domain1\config directory and look for the code block shown below: `<jmx-connector accept-all="false" address="0.0.0.0" auth-realm-name="admin-realm" enabled="true" name="system" port="8686" protocol="rmi_jrmp" security-enabled="false">` The user name specified against the auth-realm-name parameter in the code block above can be configured as the JMX User, and the Password of that user can be specified against JMX Password. Confirm the JMX Password you specify by retyping that password in the Confirm Password text box.
JNDIName	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 Provider	This test uses a JMX Provider to access the MBean attributes of the GlassFish Enterprise 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 GlassFish Enterprise 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
Create method calls	Indicates the number of times the create method was invoked on this container.	Number	
Ready method calls	Indicates the number of times in the last measurement period the 'Ready' method was invoked on this container.	Number	A bean in the cache represents the ready state in the EJB lifecycle. This means that the bean has an identity (for example, a primary key or session ID) associated with it.

Measurement	Description	Measurement Unit	Interpretation
Beans in passivate state	Indicates the number of beans in this container that were in the passivate state during the last measurement period.	Number	Beans moving out of the cache have to be passivated or destroyed according to the EJB lifecycle. Once passivated, a bean has to be activated to come back into the cache. Typically, bean passivation occurs when the max-cache-size setting is exceeded and the cache overflows. A bean is also passivated if it outlives the cache-idle-timeout-in-seconds settings.
Remove method calls	Indicates the number of number of times the 'remote' method was invoked on the beans in this container during the last measurement period.	Number	If a stateful session bean remains idle in the backup store for a duration beyond the configured removal-timeout-in-seconds, then it is removed from the backup store and will not be accessible to the client. The default value is 60 minutes.

3.4.5 GlassFish EJB Stateless Session Bean Statistics Test

The stateless container manages stateless session beans, which, by definition, do not carry client-specific states. All session beans (of a particular type) are considered equal. A stateless session bean container uses a bean pool to service requests.

This test monitors the stateless containers and reports the number and nature of methods that were invoked on each container. In the process, the test reports the status of the stateless session beans - whether they have just been created, or are in the 'ready' state, in the 'passivate' state, or have been removed from the container.

Target of the test : A GlassFish server

Agent deploying the test : An internal/remote agent

Outputs of the test : One set of results for each stateless session bean container configured on the GlassFish Enterprise server being monitored.

Configurable parameters for the test

Parameters	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
JMX Remote Port	To collect metrics from a GlassFish Enterprise server, the eG agent monitoring that server should be configured to use **JMX** to connect to the JRE used by the server and pull out the metrics of interest. By default, **JMX** support is enabled for the JRE used by the GlassFish Enterprise Server. The JMX connector listens on port 8686 by default. Therefore, type 8686 as the JMX Remote Port. If JMX listens on a different port in your environment, then specify the same here. To know the port at which JMX listens, open the **domain.xml** file in the <GLASSFISH_INSTALL_DIR>\Sun\AppServer\domains\domain1\config directory and look for the code block shown below: `<jmx-connector accept-all="false" address="0.0.0.0" auth-realm-name="admin-realm" enabled="true" name="system" port="8686" protocol="rmi_jrmp" security-enabled="false">` The port number specified against the Port parameter in the code block above refers to the JMX Remote Port.
JMX User, JMX Password, and Confirm Password	By default, the JMX connector on the GlassFish Enterprise Server requires authentication. To enable the eG agent to use JMX therefore, you need to configure the agent with the credentials of a user who is authorized to use JMX. Typically, the GlassFish administrator has the right to use JMX. You can hence configure the JMX User and JMX Password parameters with the credentials of the *administrator*. However, if you prefer not to expose the credentials of an administrator owing to security considerations, you can use the credentials of any other user with access rights to JMX. To know the name of such a user, open the **domain.xml** file in the <GLASSFISH_INSTALL_DIR>\Sun\AppServer\domains\domain1\config directory and look for the code block shown below: `<jmx-connector accept-all="false" address="0.0.0.0" auth-realm-name="admin-realm" enabled="true" name="system" port="8686" protocol="rmi_jrmp" security-enabled="false">` The user name specified against the auth-realm-name parameter in the code block above can be configured as the JMX User, and the Password of that user can be specified against JMX Password. Confirm the JMX Password you specify by retyping that password in the Confirm Password text box.
JNDIName	The JNDIName is a lookup name for connecting to the JMX connector. By default, this

Parameters	Description
	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 Provider	This test uses a JMX Provider to access the MBean attributes of the GlassFish Enterprise 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 GlassFish Enterprise 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
Create method calls	Indicates the number of times the create method was invoked on this container.	Number	
Ready method calls	Indicates the number of times in the last measurement period the 'Ready' method was invoked on this container.	Number	
Remove method calls	Indicates the number of number of times the 'remote' method was invoked on the beans in this container during the last measurement period.	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.