Monitoring Tuxedo Domain Server

eG Innovations Product Documentation

www.eginnovations.com

Total Performance Visibility

Table of Contents

Table of Figures

Chapter 1: Introduction

The BEA Tuxedo system is a middleware product that distributes applications across multiple platforms, databases, and operating systems using message-based communications and, if desired, distributed transaction processing. Middleware is used with client/server applications to distribute processing among multiple servers, manage distributed transactions, and integrate multiple database platforms. Middleware systems are sometimes known as “on-line transaction processing” or “OLTP” systems.

The BEA Tuxedo system provides the following:

- An industry standard for the creation and central administration of distributed on- line transaction applications in a heterogeneous client/server environment.
- Ease of use for application developers, who do not need to know all the details about server locations, routing, or platforms used. In a BEA Tuxedo application, these aspects of a program are transparent.
- The fundamental underpinnings for creating, managing, and maintaining reliable, high performance, easily managed distributed systems.

These powerful capabilities have made the BEA Tuxedo system a key component in IT infrastructures delivering mission-critical services to end-users. A system malfunction can therefore, delay service delivery, and cause the business to suffer colossal losses. To avoid such repercussions, it is best to monitor the performance of the Tuxedo domain server on a continuous basis. This is exactly what eG Enterprise does!

Chapter 2: How to Monitor Tuxedo Domain Server using eG Enterprise?

eG Enterprise provides two specialized monitoring models for the Tuxedo Domain Server - a Tuxedo Domain monitoring model and a Tuxedo monitoring model. The Tuxedo Domain monitoring model is capable of monitoring the Tuxedo Domain server on Windows and Solaris platforms only. Whereas, the Tuxedo monitoring model can pull out metrics from any Tuxedo Domain server, regardless of the operating system on which it executes. In other words, using the Tuxedo model you can monitor the health of a Tuxedo Domain server on a Windows, Linux, Solaris, AIX, or HPUX platform.

Both the Tuxedo Domain and Tuxedo monitoring models are agent-based models only.

This document discusses both these models in detail.

2.1 Managing the Tuxedo Domain Server

The Tuxedo domain server can be automatically discovered by eG Enterprise. However, it can be added manually using the eG administrative interface. Remember that the eG Enterprise automatically manages the components that are added manually. Discovered components, however, are managed using the **COMPONENTS - MANAGE/UNMANAGE** page. To manually add the Tuxedo Domain server component, do the following:

1. Log into the eG administrative interface.
2. If a Tuxedo domain server is already discovered, , then directly proceed towards managing it using the **COMPONENTS - MANAGE/UNMANAGE** page (Infrastructure -> Components -> Manage/Unmanage).
3. If the Tuxedo domain server is yet to be discovered, then run discovery (Infrastructure -> Components -> Discovery) to get it discovered or add the Tuxedo domain server manually using the **COMPONENTS** page (Infrastructure -> Components -> Add/Modify). Figure 2.1 clearly illustrates the process of adding a Tuxedo domain server.

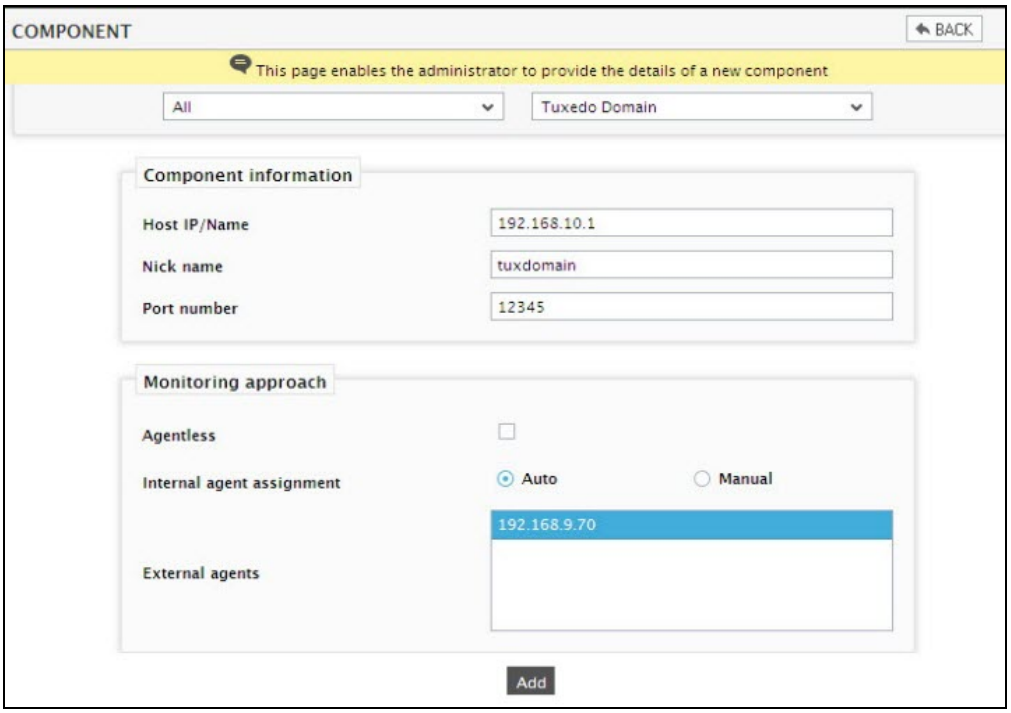

Figure 2.1: Adding a component of type Tuxedo Domain server

3. Specify the **Host IP/Name** and the **Nick name** of the Tuxedo Domain server in Chapter 2. Then, click the **Add** button to register the changes.
4. Next, try to sign out of the eG administrative interface. Upon doing so, a list of unconfigured tests will appear as in Figure 2.2 prompting you to configure the tests pertaining to the Tuxedo Domain server.

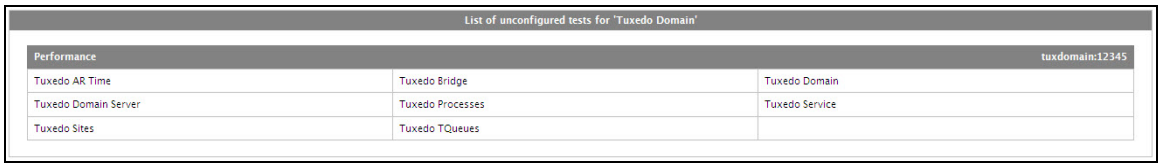

Figure 2.2: A list of tests to be configured

5. Click on any test in the list of unconfigured tests. For instance, click on the **Tuxedo Services** test to configure it. In the page that appears, specify the parameters as shown Figure 2.3.

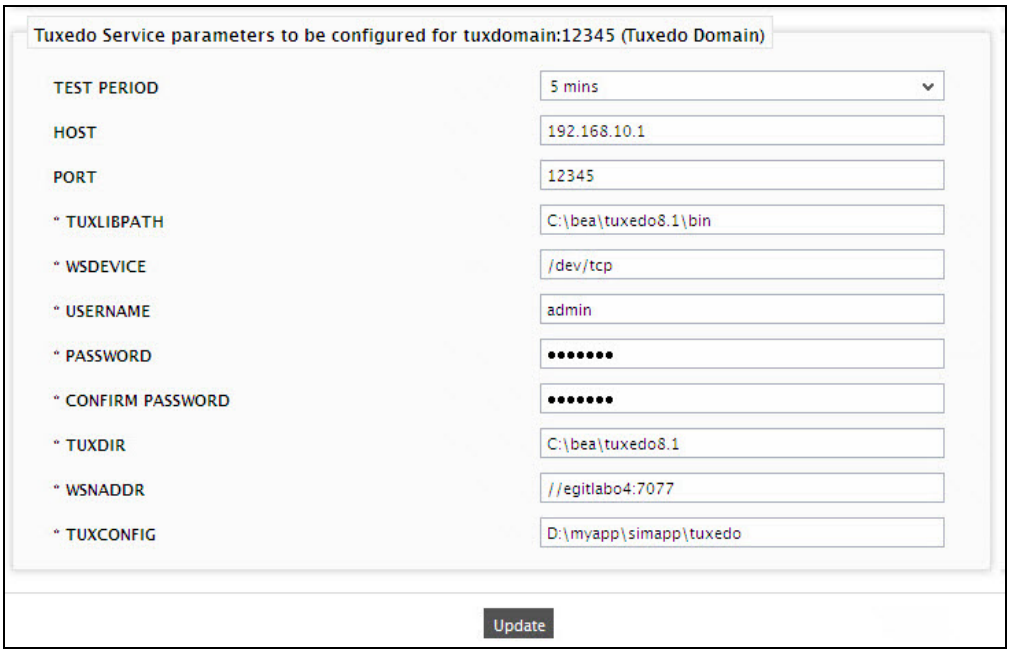

Figure 2.3: Configuring the Tuxedo Services test

6. To know how to configure the parameters, refer to the Section **3.3.1**.
7. Finally, signout of the eG administrative interface.

Chapter 3: Monitoring the Tuxedo Domain Servers Running on Windows/Solaris Systems

Figure 3.1 depicts the Tuxedo Domain monitoring model, which focuses on the health of those Tuxedo Domain servers that run on Windows/Solaris systems. Each layer of this model is mapped to tests that use an executable that is by default bundled into the Tuxedo Domain server installation in your environment to pull out useful statistics from the server. These tests collect a plethora of performance metrics from the system, compare the actual performance with pre-set service levels, and proactively alert administrators to probable violations.

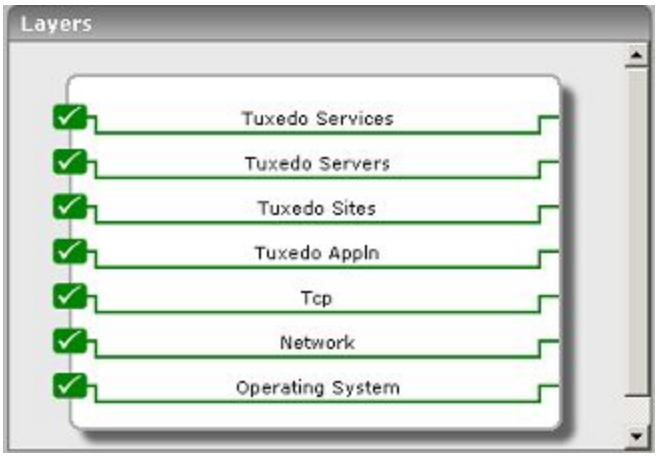

Figure 3.1: Layer model of a Tuxedo Domain server

The sections to come will elaborately discuss each of the top 4 layers of the layer model. For details on the 3 layers at the bottom of the layer model, please refer to the Monitoring Unix and Windows Servers document.

3.1 The Tuxedo Appln Layer

The tests that execute on the **Tuxedo Appln** layer indicate whether the Tuxedo domain server is available or not, and if so, how quickly it responds to client requests. In addition, the tests also report the current load on a Tuxedo application and queue statistics.

Figure 3.2: Tests executing on the Tuxedo Appln layer

3.1.1 Tuxedo AR Time Test

This test measures the availability and response time of a Tuxedo application.

Target of the test : A Tuxedo Domain Server

Agent deploying the test : An internal agent

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

Configurable parameters for the test

Parameter	Description
Test period	How often should the test be executed
Host	The host for which the test is to be configured.
WSNAddr	The network address of the workstation listener that is to be contacted for access to the Tuxedo application. This variable is required for Workstation clients and is ignored for native clients. When invoked by a workstation client, this variable is used within tpinit(). **Example:** TCP/IP addresses may be specified in the following forms: *//host.name:port_number* OR *//#.#.#.#:port_number*, where port_number is the port at which the WSL (Workstation listener) is started for a Tuxedo application or domain.
WSDevice	Specify the name of the device to be used to access the network. This variable is

Parameter	Description
	required for Workstation clients and is ignored for native clients. When invoked by a workstation client, this variable is used within tpinit(). **Example:** On Unix machines, the value of this parameter can be */dev/tcp*.
TuxLibPath	Specify the path to the Tuxedo library. **Example:** On Unix machines, the value of this parameter can be in the form: *<TUXEDO_INSTALL_DIR>/lib*. On Windows, it can be in the form: *<TUXEDO_INSTALL_DIR>\lib*.
TuxDir	Specify the installation directory of the Tuxedo domain server.
UserName	If security is enabled for an application, then provide the user name that can be used to connect to the application. Otherwise, enter *None*.
Password	If security is enabled for an application, then provide the password for the above-mentioned UserName. Otherwise, enter *None*.
Confirm Password	Confirm the password (if specified) by retyping it here.
TuxConfig	Provide the full path to the tuxconfig file used to run the Tuxedo application.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Availability	Indicates the availability of the Tuxedo application	Percent	
Response time	Indicates the response time of the Tuxedo application	Sec	

3.1.2 Tuxedo Domain Test

This test gathers the performance statistics pertaining to a Tuxedo domain (i.e a Tuxedo application).

Target of the test : A Tuxedo Domain Server

Agent deploying the test : An internal agent

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

Configurable parameters for the test

Parameter	Description
Test period	How often should the test be executed
Host	The host for which the test is to be configured.
WSNAddr	The network address of the workstation listener that is to be contacted for access to the Tuxedo application. This variable is required for Workstation clients and is ignored for native clients. When invoked by a workstation client, this variable is used within tpinit(). **Example:** TCP/IP addresses may be specified in the following forms: *//host.name:port_number* OR *//#.#.#.#:port_number*, where port_number is the port at which the WSL (Workstation listener) is started for a Tuxedo application or domain.
WSDevice	Specify the name of the device to be used to access the network. This variable is required for Workstation clients and is ignored for native clients. When invoked by a workstation client, this variable is used within tpinit(). **Example:** On Unix machines, the value of this parameter can be */dev/tcp*.
TuxLibPath	Specify the path to the Tuxedo library. **Example:** On Unix machines, the value of this parameter can be in the form: <*TUXEDO_INSTALL_DIR*>*/lib*. On Windows, it can be in the form: <*TUXEDO_INSTALL_DIR*>*\lib*.
TuxDir	Specify the installation directory of the Tuxedo domain server.
UserName	If security is enabled for an application, then provide the user name that can be used to connect to the application. Otherwise, enter *None*.
Password	If security is enabled for an application, then provide the password for the above-mentioned UserName. Otherwise, enter *None*.
Confirm Password	Confirm the password (if specified) by retyping it here.
TuxConfig	Provide the full path to the tuxconfig file used to run the Tuxedo application.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Current machines in domain	Indicates the number of machines currently running in a Tuxedo application	Number	
Current servers in domain	Indicates the number of servers available currently in a Tuxedo application	Number	
Current queues	Indicates the number of queues currently existing within a Tuxedo application	Number	
Current services in domain	Indicates the number of services currently available in a Tuxedo application	Number	
Machine utilization	Indicates the percentage of configured machines that have been utilized in a Tuxedo application	Percent	
Server utilization	Indicates the percentage of configured servers utilized in a Tuxedo application	Percent	
Queue utilization	Indicates the percentage of queues used	Percent	
Service utilization	Indicates the percentage of configured services used in a Tuxedo application	Percent	

3.1.3 Tuxedo TQueues Test

This test reports measurements pertaining to queues in a Tuxedo application.

Target of the test : A Tuxedo Domain Server

Agent deploying the test : An internal agent

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

Configurable parameters for the test

Parameter	Description
Test period	How often should the test be executed
Host	The host for which the test is to be configured.
WSNAddr	The network address of the workstation listener that is to be contacted for access to the Tuxedo application. This variable is required for Workstation clients and is ignored for native clients. When invoked by a workstation client, this variable is used within tpinit(). **Example:** TCP/IP addresses may be specified in the following forms: *//host.name:port_number* OR *//#.#.#.#:port_number*, where port_number is the port at which the WSL (Workstation listener) is started for a Tuxedo application or domain.
WSDevice	Specify the name of the device to be used to access the network. This variable is required for Workstation clients and is ignored for native clients. When invoked by a workstation client, this variable is used within tpinit(). **Example:** On Unix machines, the value of this parameter can be */dev/tcp*.
TuxLibPath	Specify the path to the Tuxedo library. **Example:** On Unix machines, the value of this parameter can be in the form: <*TUXEDO_INSTALL_DIR*>*/lib*. On Windows, it can be in the form: <*TUXEDO_INSTALL_DIR*>*\lib*.
TuxDir	Specify the installation directory of the Tuxedo domain server.
UserName	If security is enabled for an application, then provide the user name that can be used to connect to the application. Otherwise, enter *None*.
Password	If security is enabled for an application, then provide the password for the above-mentioned UserName. Otherwise, enter *None*.
Confirm Password	Confirm the password (if specified) by retyping it here.
TuxConfig	Provide the full path to the tuxconfig file used to run the Tuxedo application.

Measurements made by the test

<table>
<tr><th>Measurement</th><th>Description</th><th>Measurement Unit</th><th>Interpretation</th></tr>
<tr><td>Queue state</td><td>Indicates the state of the queue.</td><td>Number</td><td>It could take any of the following values:
<table>
<tr><th>Value</th><th>State</th><th>Explanation</th></tr>
<tr><td>1</td><td>Active</td><td>Indicates that at least one server associated with the queue object is active</td></tr>
<tr><td>2</td><td>Migrating</td><td>Indicates that the server (s) associated with the queue object are currently in the migrating state</td></tr>
<tr><td>3</td><td>Suspended</td><td>Indicates that the server (s) associated with the queue object are currently in the</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Measurement</th><th>Description</th><th>Measurement Unit</th><th>Interpretation</th></tr>
<tr><td></td><td></td><td></td><td>
<table>
<tr><th>Value</th><th>State</th><th>Explanation</th></tr>
<tr><td></td><td></td><td>suspended state</td></tr>
<tr><td>4</td><td>Partitioned</td><td>Indicates that the server (s) associated with the queue object are currently in the partitioned state</td></tr>
</table>
</td></tr>
<tr><td>Server count</td><td>Indicates the number of active servers associated with the queue</td><td>Number</td><td></td></tr>
<tr><td>Request queued</td><td>Indicates the number of requests currently queued in the queue</td><td>Number</td><td></td></tr>
<tr><td>Current workload</td><td>Indicates the workload currently associated with the queue</td><td>Number</td><td></td></tr>
</table>

The Tuxedo Sites Layer

Figure 3.3 below lists the tests that run on this layer. These tests monitor the performance of every site configured on the Tuxedo domain server and the bridges between the sites.

Figure 3.3: Figure 2.3: Tests executing on the Tuxedo Sites layer

3.1.4 Tuxedo Bridges Test

This test reports statistics pertaining to the bridges between the sites in a Tuxedo application.

Target of the test : A Tuxedo Domain Server

Agent deploying the test : An internal agent

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

Configurable parameters for the test

Parameter	Description
Test period	How often should the test be executed.
Host	The host for which the test is to be configured.
WSNAddr	The network address of the workstation listener that is to be contacted for access to the Tuxedo application. This variable is required for Workstation clients and is ignored for native clients. When invoked by a workstation client, this variable is used within tpinit(). **Example:** TCP/IP addresses may be specified in the following forms: *//host.name:port_number* OR *//#.#.#.#:port_number*, where port_number is the port at which the WSL (Workstation listener) is started for a Tuxedo application or domain.
WSDevice	Specify the name of the device to be used to access the network. This variable is

Parameter	Description
	required for Workstation clients and is ignored for native clients. When invoked by a workstation client, this variable is used within tpinit(). **Example:** On Unix machines, the value of this parameter can be */dev/tcp*.
TuxLibPath	Specify the path to the Tuxedo library. **Example:** On Unix machines, the value of this parameter can be in the form: <*TUXEDO_INSTALL_DIR*>*/lib*. On Windows, it can be in the form: <*TUXEDO_INSTALL_DIR*>*\lib*.
TuxDir	Specify the installation directory of the Tuxedo domain server.
UserName	If security is enabled for an application, then provide the user name that can be used to connect to the application. Otherwise, enter *None*.
Password	If security is enabled for an application, then provide the password for the above-mentioned UserName. Otherwise, enter *None*.
Confirm Password	Confirm the password (if specified) by retyping it here.
TuxConfig	Provide the full path to the tuxconfig file used to run the Tuxedo application.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Bridge state	Indicates the state of a bridge between two sites in a Tuxedo application	Number	It could take any of the following values: **Value** | **State** | **Explanation** 1 | Active | Indicates that the bridge or connection between two sites in a Tuxedo application is

<table>
<tr><th>Measurement</th><th>Description</th><th>Measurement Unit</th><th>Interpretation</th></tr>
<tr><td></td><td></td><td></td><td>
<table>
<tr><th>Value</th><th>State</th><th>Explanation</th></tr>
<tr><td></td><td></td><td>active or established</td></tr>
<tr><td>2</td><td>Suspended</td><td>Indicates that an established bridge or connection was terminated due to an error condition, and reconnection has been suspended for at least the amount of time indicated in the TA_SUSPTIME attribute value</td></tr>
<tr><td>3</td><td>Pending</td><td>Indicates that an asynchronous bridge or connection has been requested, but has not yet been completed. The final</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Measurement</th><th>Description</th><th>Measurement Unit</th><th>Interpretation</th></tr>
<tr><td></td><td></td><td></td><td>
<table>
<tr><th>Value</th><th>State</th><th>Explanation</th></tr>
<tr><td></td><td></td><td>outcome of the connection request has not been determined.</td></tr>
<tr><td>4</td><td>Inactive</td><td>Indicates that the bridge or connection between two sites is inactive</td></tr>
</table>
</td></tr>
<tr><td>Data receive rate</td><td>Indicates the rate at which bytes are sent from the destination logical machine to the source logical machine</td><td>KB/sec</td><td></td></tr>
<tr><td>Data transmit rate</td><td>Indicates the rate at which bytes are sent from the source logical machine to the destination logical machine</td><td>KB/sec</td><td></td></tr>
<tr><td>Message receive rate</td><td>Indicates the rate at which messages are sent from the destination logical machine to the source logical machine</td><td>Msgs/sec</td><td></td></tr>
<tr><td>Message transmit rate</td><td>Indicates the rate at which messages are sent from the source logical machine to the</td><td>Msgs/sec</td><td></td></tr>
</table>

Measurement	Description	Measurement Unit	Interpretation
	destination logical machine		

3.1.5 Tuxedo Sites Test

This test reports statistics pertaining to each site configured under a Tuxedo domain.

Target of the test : A Tuxedo Domain Server

Agent deploying the test : An internal agent

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

Configurable parameters for the test

Parameter	Description
Test period	How often should the test be executed
Host	The host for which the test is to be configured.
WSNAddr	The network address of the workstation listener that is to be contacted for access to the Tuxedo application. This variable is required for Workstation clients and is ignored for native clients. When invoked by a workstation client, this variable is used within tpinit(). **Example:** TCP/IP addresses may be specified in the following forms: *//host.name:port_number* OR *//#.#.#.#:port_number*, where port_number is the port at which the WSL (Workstation listener) is started for a Tuxedo application or domain.
WSDevice	Specify the name of the device to be used to access the network. This variable is required for Workstation clients and is ignored for native clients. When invoked by a workstation client, this variable is used within tpinit(). **Example:** On Unix machines, the value of this parameter can be */dev/tcp*.
TuxLibPath	Specify the path to the Tuxedo library.

Parameter	Description
	Example: On Unix machines, the value of this parameter can be in the form: <*TUXEDO_INSTALL_DIR*>*/lib*. On Windows, it can be in the form: <*TUXEDO_INSTALL_DIR*>*\lib*.
TuxDir	Specify the installation directory of the Tuxedo domain server.
UserName	If security is enabled for an application, then provide the user name that can be used to connect to the application. Otherwise, enter *None*.
Password	If security is enabled for an application, then provide the password for the above-mentioned UserName. Otherwise, enter *None*.
Confirm Password	Confirm the password (if specified) by retyping it here.
TuxConfig	Provide the full path to the tuxconfig file used to run the Tuxedo application.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Site state	Indicates the state of a site configured under a Tuxedo domain.		It could take any of the following values: **Value** / **State** / **Explanation** 1 / Active / Indicates that the site is defined and active 2 / Partitioned / Indicates that the site defined and active, but is currently unreachable. This state is equivalent to Active for the

<table>
<tr><th>Measurement</th><th>Description</th><th>Measurement Unit</th><th>Interpretation</th></tr>
<tr><td></td><td></td><td></td><td>
<table>
<tr><th>Value</th><th>State</th><th>Explanation</th></tr>
<tr><td></td><td></td><td>purpose of determining permissions.</td></tr>
<tr><td>3</td><td>Inactive</td><td>Indicates that the site is defined and inactive</td></tr>
</table>
</td></tr>
<tr><td>Transaction aborts</td><td>Indicates the number of clients, both native and Workstation, who are currently logged in to the site</td><td>Number</td><td></td></tr>
<tr><td>Transaction commits</td><td>Indicates the number of Workstation clients, currently logged in to the site</td><td>Number</td><td></td></tr>
<tr><td>TListen state</td><td>Indicates the state of the Tlisten process running on this site</td><td>Number</td><td>This measure takes the value 1 or 2. 1 implies that the process is active and 2 implies that the process is not running or inactive.</td></tr>
<tr><td>Total clients</td><td>Indicates the number of clients, both native and Workstation, who are currently logged in to the site</td><td>Number</td><td></td></tr>
<tr><td>Workstation clients</td><td>Indicates the number of Workstation clients, currently logged in to the site</td><td>Number</td><td></td></tr>
<tr><td>Conversations</td><td>Indicates the number of active conversations with participants on the</td><td>Number</td><td></td></tr>
</table>

Measurement	Description	Measurement Unit	Interpretation
	site		
Global transactions	Indicates the number of global transactions the site is currently involved in	Number	
Current load	Indicates the rate at which load is queued on this site	Loads/sec	
Work completions	Indicates the rate at which work is completed by this site	Works/sec	
Work initiations	Indicates the rate at which work is initiated on this site by clients/servers	Works/sec	
Transaction initiations	Indicates the rate at which transaction is initiated from this site	Trans/sec	

3.2 The Tuxedo Servers Layer

This layer generates performance statistics pertaining to the servers and processes running on a Tuxedo application (see Figure 2.4).

Figure 3.4: Tests associated with the Tuxedo Servers layer

3.2.1 TuxedoServer Test

This test measures the performance of each server running under each site hosted on a Tuxedo application.

Target of the test : A Tuxedo Domain Server

Agent deploying the test : An internal agent

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

Configurable parameters for the test

Parameter	Description
Test period	How often should the test be executed
Host	The host for which the test is to be configured.
WSNAddr	The network address of the workstation listener that is to be contacted for access to the Tuxedo application. This variable is required for Workstation clients and is ignored for native clients. When invoked by a workstation client, this variable is used within tpinit(). **Example:** TCP/IP addresses may be specified in the following forms: *//host.name:port_number* OR *//#.#.#.#:port_number*, where port_number is the port at which the WSL (Workstation listener) is started for a Tuxedo application or domain.
WSDevice	Specify the name of the device to be used to access the network. This variable is required for Workstation clients and is ignored for native clients. When invoked by a workstation client, this variable is used within tpinit(). **Example:** On Unix machines, the value of this parameter can be */dev/tcp*.
TuxLibPath	Specify the path to the Tuxedo library. **Example:** On Unix machines, the value of this parameter can be in the form: *<TUXEDO_INSTALL_DIR>/lib*. On Windows, it can be in the form: *<TUXEDO_INSTALL_DIR>\lib*.

Parameter	Description
TuxDir	Specify the installation directory of the Tuxedo domain server.
UserName	If security is enabled for an application, then provide the user name that can be used to connect to the application. Otherwise, enter *None*.
Password	If security is enabled for an application, then provide the password for the above-mentioned UserName. Otherwise, enter *None*.
Confirm Password	Confirm the password (if specified) by retyping it here.
TuxConfig	Provide the full path to the tuxconfig file used to run the Tuxedo application.

Measurements made by the test

<table>
<tr><th>Measurement</th><th>Description</th><th>Measurement Unit</th><th>Interpretation</th></tr>
<tr><td>Server state</td><td>Indicates the state of a server running on a particular site.</td><td>Number</td><td>It could take any of the following values:
<table>
<tr><th>Value</th><th>State</th><th>Explanation</th></tr>
<tr><td>1</td><td>Active</td><td>Indicates that the server object is defined and active</td></tr>
<tr><td>2</td><td>Migrating</td><td>Indicates that the server is defined and currently in a state of migration to the server group’s secondary logical machine</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Measurement</th><th>Description</th><th>Measurement Unit</th><th>Interpretation</th></tr>
<tr><td></td><td></td><td></td><td>
<table>
<tr><th>Value</th><th>State</th><th>Explanation</th></tr>
<tr><td>3</td><td>Cleaning</td><td>Indicates that the server object is defined and is currently being cleaned up by the system after an abnormal death</td></tr>
<tr><td>4</td><td>Restarting</td><td>Indicates that the server object defined and is currently being restarted by the system after an abnormal death</td></tr>
<tr><td>5</td><td>Suspended</td><td>Indicates that the server object is defined and is currently suspended</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Measurement</th><th>Description</th><th>Measurement Unit</th><th>Interpretation</th></tr>
<tr><td></td><td></td><td></td><td>
<table>
<tr><th>Value</th><th>State</th><th>Explanation</th></tr>
<tr><td></td><td></td><td>pending shutdown</td></tr>
<tr><td>6</td><td>Partitioned</td><td>Indicates that the server object is defined and active; however, the machine where the server is running is currently partitioned from the master site</td></tr>
<tr><td>7</td><td>Dead</td><td>Indicates that the server object is defined, identified as active in the bulletin board, but currently not running due to an abnormal death</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Measurement</th><th>Description</th><th>Measurement Unit</th><th>Interpretation</th></tr>
<tr><td></td><td></td><td></td><td>
<table>
<tr><th>Value</th><th>State</th><th>Explanation</th></tr>
<tr><td>8</td><td>Inactive</td><td>Indicates that the server object is defined and inactive</td></tr>
</table>
</td></tr>
<tr><td>Workload completions</td><td>Indicates the rate at which workload was completed by the server.</td><td>Works/Sec</td><td></td></tr>
<tr><td>Total conversations</td><td>Indicates the total number of conversations initiated by the server</td><td>Number</td><td></td></tr>
<tr><td>Active conversations</td><td>Indicates the number of currently active conversations initiated by the server</td><td>Number</td><td></td></tr>
<tr><td>Total requests</td><td>Indicates the total number of requests made by the server via tpcall() or tpacall().</td><td>Number</td><td></td></tr>
<tr><td>Active requests</td><td>Indicates the number of currently active requests made by the server via tpcall () or tpacall().</td><td>Number</td><td></td></tr>
<tr><td>Transaction initiations</td><td>Indicates the rate at which transactions were begun by the</td><td>Trans/sec</td><td></td></tr>
</table>

Measurement	Description	Measurement Unit	Interpretation
	server since it was last restarted.		
Transaction aborts	Indicates the rate at which transactions were aborted by the server since it was last restarted.	Trans/sec	
Transaction commits	Indicates the rate at which transactions were committed by the server since it was last restarted.	Trans/Sec	

3.2.2 Tuxedo Processes Test

This test reports statistics related to the processes running on a site hosted on a Tuxedo application.

Target of the test : A Tuxedo Domain Server

Agent deploying the test : An internal agent

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

Configurable parameters for the test

Parameter	Description
Test period	How often should the test be executed
Host	The host for which the test is to be configured.
WSNAddr	The network address of the workstation listener that is to be contacted for access to the Tuxedo application. This variable is required for Workstation clients and is ignored for native clients. When invoked by a workstation client, this variable is used within tpinit(). **Example:** TCP/IP addresses may be specified in the following forms: *//host.name:port_number*

Parameter	Description
	OR *//#.#.#.#:port_number*, where port_number is the port at which the WSL (Workstation listener) is started for a Tuxedo application or domain.
WSDevice	Specify the name of the device to be used to access the network. This variable is required for Workstation clients and is ignored for native clients. When invoked by a workstation client, this variable is used within tpinit(). **Example:** On Unix machines, the value of this parameter can be */dev/tcp*.
TuxLibPath	Specify the path to the Tuxedo library. **Example:** On Unix machines, the value of this parameter can be in the form: *<TUXEDO_INSTALL_DIR>/lib*. On Windows, it can be in the form: *<TUXEDO_INSTALL_DIR>\lib*.
TuxDir	Specify the installation directory of the Tuxedo domain server.
UserName	If security is enabled for an application, then provide the user name that can be used to connect to the application. Otherwise, enter *None*.
Password	If security is enabled for an application, then provide the password for the above-mentioned UserName. Otherwise, enter *None*.
Confirm Password	Confirm the password (if specified) by retyping it here.
TuxConfig	Provide the full path to the tuxconfig file used to run the Tuxedo application.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Processes running	Indicates the number of processes running on a Tuxedo application	Number	
CPU utilization	Indicates the percentage of CPU utilized by a process	Percent	
Memory utilization	Indicates the percentage of memory utilized by a process	Percent	

3.3 The Tuxedo Services Layer

This layer tracks the health of the services running on the Tuxedo domain.

Figure 3.5: Tests that execute on the Tuxedo Services layer

3.3.1 Tuxedo Service Test

This test measures the performance of services provided by servers running on a Tuxedo application.

Target of the test : A Tuxedo Domain Server

Agent deploying the test : An internal agent

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

Configurable parameters for the test

Parameter	Description
Test period	How often should the test be executed.
Host	The host for which the test is to be configured.
WSNAddr	The network address of the workstation listener that is to be contacted for access to the Tuxedo application. This variable is required for Workstation clients and is ignored for native clients. When invoked by a workstation client, this variable is used within tpinit(). **Example:** TCP/IP addresses may be specified in the following forms:

Parameter	Description
	//host.name:port_number OR *//#.#.#.#:port_number*, where port_number is the port at which the WSL (Workstation listener) is started for a Tuxedo application or domain.
WSDevice	Specify the name of the device to be used to access the network. This variable is required for Workstation clients and is ignored for native clients. When invoked by a workstation client, this variable is used within tpinit(). **Example:** On Unix machines, the value of this parameter can be */dev/tcp*.
TuxLibPath	Specify the path to the Tuxedo library. **Example:** On Unix machines, the value of this parameter can be in the form: *<TUXEDO_INSTALL_DIR>/lib*. On Windows, it can be in the form: *<TUXEDO_INSTALL_DIR>\lib*.
TuxDir	Specify the installation directory of the Tuxedo domain server.
UserName	If security is enabled for an application, then provide the user name that can be used to connect to the application. Otherwise, enter *None*.
Password	If security is enabled for an application, then provide the password for the above-mentioned UserName. Otherwise, enter *None*.
Confirm Password	Confirm the password (if specified) by retyping it here.
TuxConfig	Provide the full path to the tuxconfig file used to run the Tuxedo application.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Service state	Indicates the state of a service provided by a server in a Tuxedo domain.	Number	It could take any of the following values: Value | State | Explanation 1 | Active | Indicates

Measurement	Description	Measurement Unit	Interpretation

Value	State	Explanation
		that the service object is defined and active
2	Suspended	Indicates that the service object is defined and is currently suspended. The service is not available for access by the application in this state
3	Partitioned	Indicates that the service object is defined and active, but is currently partitioned from the master site
4	Inactive	Indicates that the service

Measurement	Description	Measurement Unit	Interpretation
			Value: (blank) | State: (blank) | Explanation: object is defined and inactive
Load imposed	Indicates the load imposed by the service object on the system.	Number	Service loads are used for load balancing purposes. This means that queues with higher enqueued workloads are less likely to be chosen for a new request.
Requests completed	Indicates the rate at which requests are completed by the service object.	Reqs/Sec	
Requests queued	Indicates the number of requests currently enqueued to the service.	Number	

3.4 Troubleshooting

If the tests associated with the Tuxedo domain server are in **UNKNOWN** state, it could indicate any/all of the following:

1. **The eG agent has not been started.**

   If the eG agent executing the Tuxedo tests has not been started, then, the tests will not get executed. In such a case, proceed to start the eG agent. Refer to the *eG Installation Manual* for the procedure for starting an eG agent on Windows and Unix environments.

2. **The Tuxedo domain server being monitored has not been started**

   If the eG agent has already been started, then verify whether the Tuxedo domain server has been started or not. To check this, perform the following steps:

3. **In a Windows installation of the Tuxedo server, set the following environment variables for the application being monitored:**

```
set TUXDIR=pathname_of_BEA_Tuxedo_product_directory
set APPDIR=pathname_of_BEA_Tuxedo_application_directory
set TUXCONFIG=pathname_of_TUXCONFIG_file
set WEBJAVADIR=%TUXDIR%\udataobj\webgui\java
set PATH=%APPDIR%;%TUXDIR%\bin;\bin;%PATH%
```

Examples of **TUXDIR**, **APPDIR**, and **TUXCONFIG** are:

TUXDIR=C:\bea\tuxedo8.1

APPDIR=C:\home\me\simpapp

TUXCONFIG=%APPDIR%\tuxconfig

4. On a Unix server machine, set and export the following environment variables for the application being monitored:

```
TUXDIR=pathname_of_BEA_Tuxedo_product_directory
APPDIR=pathname_of_BEA_Tuxedo_application_directory
TUXCONFIG=pathname_of_TUXCONFIG_file
WEBJAVADIR=$TUXDIR/udataobj/webgui/java
PATH=$APPDIR:$TUXDIR/bin:/bin:$PATH
LD_LIBRARY_PATH=$APPDIR:$TUXDIR/lib:/lib:/usr/lib:$LD_LIBRARY_PATH
export TUXDIR APPDIR TUXCONFIG WEBJAVADIR PATH LD_LIBRARY_PATH
```

Note:

For HP-UX systems only, use **SHLIB_PATH** instead of **LD_LIBRARY_PATH**.

Examples of TUXDIR, APPDIR, and TUXCONFIG are:

TUXDIR=/home/bea/tuxedo8.1

APPDIR=/home/me/simpapp

TUXCONFIG=$APPDIR/tuxconfig

The TUXDIR, APPDIR, and TUXCONFIG environment variables (in both Windows and Unix) must match the values of the TUXDIR, APPDIR, and TUXCONFIG parameters in the

MACHINES section of the UBBCONFIG file of the corresponding application. This config file will be present in the APPDIR of the application being monitored.

5. After setting the variables properly, start the Tuxedo server by executing the following command from the command prompt:

   tmboot –y

   If the Tuxedo server has been started for the first time, then the following output will appear:

```
Booting all admin and server processes in C:\home\me\atmi\tuxconfig
INFO: BEA Tuxedo(r) System Release 8.1
INFO: Serial #: 000102-9125503751, Maxusers 25
Booting admin processes ...
exec BBL -A:
process id=24180 ... Started.
Booting server processes ...
exec simpserv -A :
process id=24181 ... Started.
2 processes started.
```

   If the Tuxedo server had already been started, then a message to that effect will appear.

6. If the Tuxedo domain server has already been started, then it could indicate that the Tuxedo tests have not been configured properly.

   While configuring Tuxedo tests, ensure that the TuxConfig and TuxDir parameters contain the same value as the TUXCONFIG and TUXDIR environment variables set earlier. In Unix environments, make sure that the value of the TuxLibPath parameter matches with the value of the environment variable ld_library_path.

Chapter 4: Platform-Independent Monitoring of Tuxedo Domain Servers

Figure 4.1 depicts the Tuxedo monitoring model, which is capable of extracting performance metrics from Tuxedo Domain servers executing on any Windows or Unix platform.

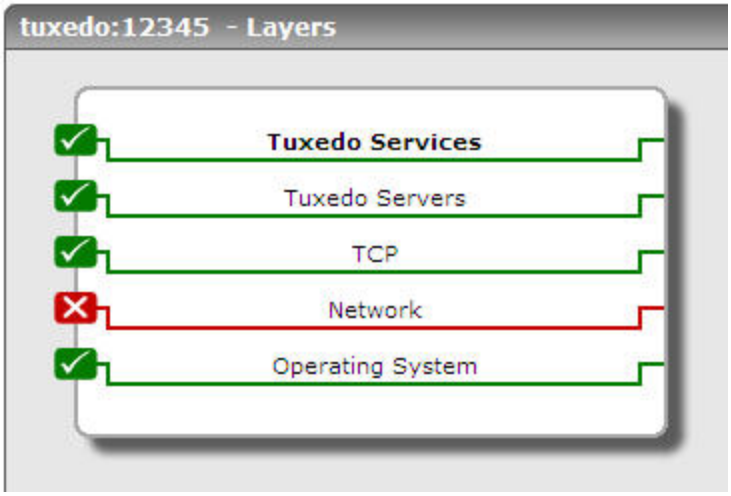

Figure 4.1: The Tuxedo monitoring model

Each layer of this model is mapped to tests that executes platform-independent API commands on the Tuxedo Domain server to pull out useful statistics from the server. You can use these metrics to answer the following performance queries related to the Tuxedo Domain server:

- Does the database connection pool have adequate free connections?
- Do you need to resize the database connection pool?
- Are there any idle/inactive administration servers? If so, which ones?
- Is any administration server overloaded?
- Have too many transactions to the Tuxedo Domain server aborted?
- Is any service unavailable on the server?
- Are any services overloaded with requests?
- How many servers, services and interfaces have registered with the Bulletin Board? How many of these are currently available?
- Are there any idle users to the server? If so, which client has the user logged in from?

As the three layers at the bottom of layer model have already been dealt with the *Monitoring Unix and Windows Servers* document, the sections that follow wil discuss the top two layers only.

4.1 The Tuxedo Servers Layer

Use the tests mapped to this layer to know how well the database connection pool has been utilized and to determine the state of each of the administration servers on the Tuxedo Domain server.

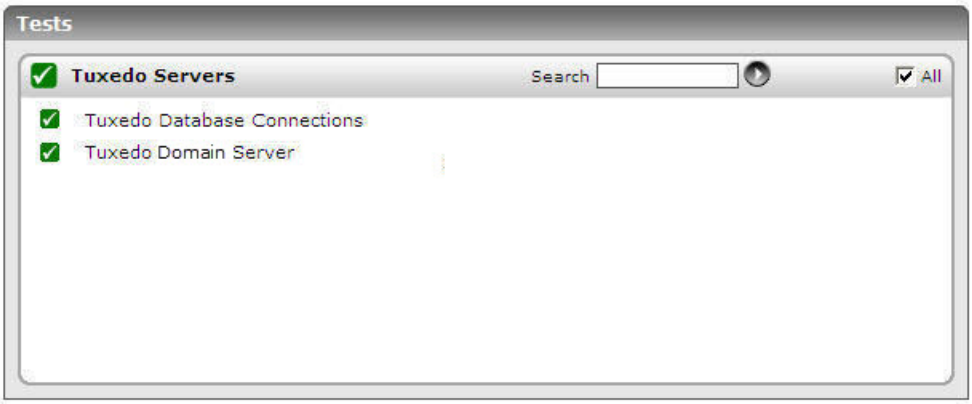

Figure 4.2: The tests mapped to the Tuxedo Servers layer

4.1.1 Tuxedo Database Connections Test

If the Tuxedo Domain server is unable to connect to the database owing to the absence of adequate connections in the connection pool, then critical server operations may fail. Using this test, you can periodically monitor the usage of each connection pool and promptly detect when a pool runs short of connections. This way, the test provides useful pointers to resizing your connection pools.

Target of the test : A Tuxedo Domain Server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for every *Group_name:Pool_name* auto-discovered from the Tuxedo server being monitored.

Configurable parameters for the test

Parameter	Description
Test period	How often should the test be executed.
Host	The host for which the test is to be configured.
Port	The port at which the specified host listens. The default port is *12345*.

Parameter	Description
TuxDir	Specify the full path to the installation directory of the Tuxedo Domain server. For example, while monitoring a Tuxedo server on Windows, your TuxDir can be: *E:\oraclehome\tuxedo11gR1*
AppDir	Specify the full path to the application directory of the Tuxedo Domain server. For example, while monitoring a Tuxedo server on Windows, your appdir can be: *E:\oraclehome\tuxedo11gR1\samples\atmi\simpapp*
TuxConfig	Each BEA Tuxedo domain is controlled by a configuration file in which installation-dependent parameters are defined. The binary version of this configuration file is called TuxConfig. The TUXCONFIG file may be given any name; the actual name is the device or system filename specified in the TUXCONFIG environment variable. Specify the name of the TuxConfig file here. For example, while monitoring a Tuxedo server on Windows, your TuxConfig can be:*E:\oraclehome\tuxedo11gR1\samples\atmi\simpapp\tuxconfig*

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Max connection pool size	Indicates the maximum number of connections in this pool.	Number	
Current connection pool size	Indicates the current size of this pool.	Number	
Used connections	Indicates the number of used connections relative to the size of this pool.	Number	Ideally, the value of this measure should be low.
Free connections	Indicates the percentage of unused connections in this pool.	Percent	A high value is desired for this measure. If the value is low, it indicates abnormal usage of the connection pool. You may want to consider resizing the pool, so that sufficient connections are always available in the pool.

4.1.2 Tuxedo Domain Server Test

An administration server is a software program that performs administration functions. Each Tuxedo managed node has the following administration servers:

- BRIDGE: An administration server that establishes the machine's listening address.
- Bulletin Board Liaison (BBL): An administration server that creates the shared memory Bulletin Board. Each master machine also has a Distinguished Bulletin Board Liaison (DBBL), which is an administration server that manages the updates to the Bulletin Board.

A Tuxedo managed node may have additional administration servers, such as Transaction Management Server (TMS), which handles transaction completion.

Failure of a critical administration server such as BBL can cause the activation of other administration servers on that machine to fail. If the DBBL fails to activate, the entire application's activation process will fail. It is hence imperative to continuously track the state of each administration server.

This test reports the current state of and the load on every administration server on the target Tuxedo server, so that server failures and improper load distribution amongst administration servers (i.e., services) are detected early and fixed.

Target of the test : A Tuxedo Domain Server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for every *Servername:ServerID* associated with each administration server on the Tuxedo Domain server being monitored.

Configurable parameters for the test

Parameter	Description
Test period	How often should the test be executed
Host	The host for which the test is to be configured.
Port	The port at which the specified host listens. The default port is *12345*.
TuxDir	Specify the full path to the installation directory of the Tuxedo Domain server. For example, while monitoring a Tuxedo server on Windows, your TuxDir can be: *E:\oraclehome\tuxedo11gR1*
AppDir	Specify the full path to the application directory of the Tuxedo Domain server. For

Parameter	Description
	example, while monitoring a Tuxedo server on Windows, your appdir can be: *E:\oraclehome\tuxedo11gR1\samples\atmi\simpapp*
TuxConfig	Each BEA Tuxedo domain is controlled by a configuration file in which installation-dependent parameters are defined. The binary version of this configuration file is called TuxConfig. The TUXCONFIG file may be given any name; the actual name is the device or system filename specified in the TUXCONFIG environment variable. Specify the name of the TuxConfig file here. For example, while monitoring a Tuxedo server on Windows, your TuxConfig can be:*E:\oraclehome\tuxedo11gR1\samples\atmi\simpapp\tuxconfig*

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Current server state	Indicates the current state of this administration server.		The values that this measure can report and their corresponding numeric equivalents have been detailed in the table below: **Measure Value** / **Numeric Value** Active: 100 Idle: 1 Inactive: 0 **Note:** By default, this measure reports the **Measure Value**s discussed in the table above to indicate the current state of a server The graph of this measure however, represents the same using the numeric equivalents only.
Requests handled	Indicates the number of requests handled by this administration server during the last measurement period.	Number	
Server load	Indicates the load on this	Number	Compare the value of this measure

Measurement	Description	Measurement Unit	Interpretation
	server during the last measurement period.		across servers to identify the server that is overloaded.

4.2 The Tuxedo Services Layer

The tests mapped to this layer measure the following:

- The number and type of transactions on the Tuxedo server
- The state of the services stored by the application servers on the Tuxedo Domain server
- The length of the queues on the Tuxedo Domain server
- The count of active conversational connections to the server
- The health of the Bulletin Board on the server;

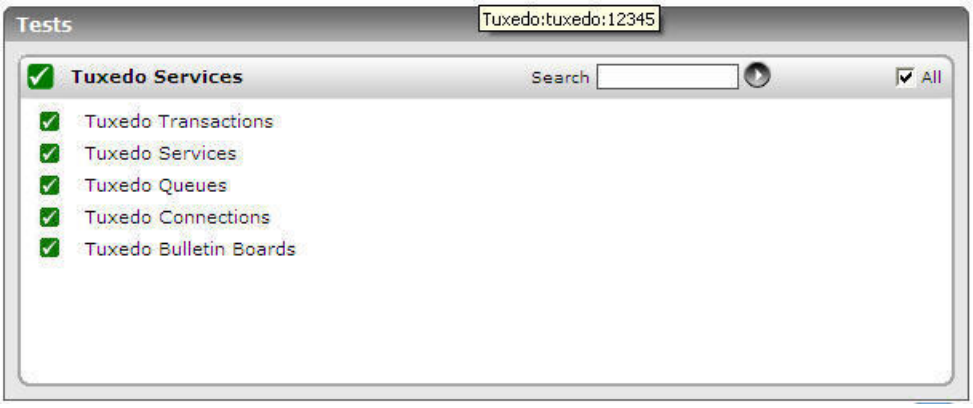

Figure 4.3: The tests mapped to the Tuxedo Services layer

4.2.1 Tuxedo Transactions Test

The BEA Tuxedo transaction management server, named TMS, is responsible for coordinating global transactions, on behalf of BEA Tuxedo applications, from their point of origin—typically on the client—across one or more server machines, and then back to the originating client. TMS tracks transaction participants and supervises a two-phase commit protocol, ensuring that transaction commit and rollback are properly handled at each site.

To measure the load on the TMS, you need to know how many transactions are being handled by the TMS and what is their current state - whether they are being committed? rolled-back? aborted?.

This can be achieved using the **Tuxedo Transactions** test. This test reports the number of transactions currently handled by TMS, and reveals the state of each transaction.

Target of the test : A Tuxedo Domain Server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for the Tuxedo Domain server being monitored.

Configurable parameters for the test

Parameter	Description
Test period	How often should the test be executed
Host	The host for which the test is to be configured.
Port	The port at which the specified host listens. The default port is *12345*.
TuxDir	Specify the full path to the installation directory of the Tuxedo Domain server. For example, while monitoring a Tuxedo server on Windows, your TuxDir can be: *E:\oraclehome\tuxedo11gR1*
AppDir	Specify the full path to the application directory of the Tuxedo Domain server. For example, while monitoring a Tuxedo server on Windows, your appdir can be: *E:\oraclehome\tuxedo11gR1\samples\atmi\simpapp*
TuxConfig	Each BEA Tuxedo domain is controlled by a configuration file in which installation-dependent parameters are defined. The binary version of this configuration file is called TuxConfig. The TUXCONFIG file may be given any name; the actual name is the device or system filename specified in the TUXCONFIG environment variable. Specify the name of the TuxConfig file here. For example, while monitoring a Tuxedo server on Windows, your TuxConfig can be:*E:\oraclehome\tuxedo11gR1\samples\atmi\simpapp\tuxconfig*
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
Transactions	Indicates the number of transactions currently handled by the server.	Number	A high value for this indicates heavy load on the TMS. To know which transactions are currently being serviced by TMS and what their status is, use the detailed diagnosis of this measure. The detailed diagnosis reveals the Global_ Transacation_Identifier, Machine_Id, Transaction_status, and Group_count. A transaction can be in of the following states: • TMGACTIVE • TMGABORTED • TMGTOBEABORTED • TMGCOMMITED • TMGCOMMITCALLED • TMGDECIDED

4.2.2 Tuxedo Services Test

An application server is a software process that stores Tuxedo services. A service is an application routine that a client can request. The non-availability of a service to clients will result in the inaccessibility of the corresponding application as well. Using this test, you can continuously track the state of each service stored by the application server and receive proactive alerts of potential service failures; this way, you can quickly isolate unavailable applications. The test also monitors the request load on each service, points you to the popular applications, and thus enables you to assess the impact of their failure.

Target of the test : A Tuxedo Domain Server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for every *group_name:program_name:service_name* associated with the each server stored by the application servers on the Tuxedo Domain server being monitored.

Configurable parameters for the test

Parameter	Description
Test period	How often should the test be executed
Host	The host for which the test is to be configured.
Port	The port at which the specified host listens. The default port is *12345*.
TuxDir	Specify the full path to the installation directory of the Tuxedo Domain server. For example, while monitoring a Tuxedo server on Windows, your TuxDir can be: *E:\oraclehome\tuxedo11gR1*
AppDir	Specify the full path to the application directory of the Tuxedo Domain server. For example, while monitoring a Tuxedo server on Windows, your appdir can be: *E:\oraclehome\tuxedo11gR1\samples\atmi\simpapp*
TuxConfig	Each BEA Tuxedo domain is controlled by a configuration file in which installation-dependent parameters are defined. The binary version of this configuration file is called TuxConfig. The TUXCONFIG file may be given any name; the actual name is the device or system filename specified in the TUXCONFIG environment variable. Specify the name of the TuxConfig file here. For example, while monitoring a Tuxedo server on Windows, your TuxConfig can be:*E:\oraclehome\tuxedo11gR1\samples\atmi\simpapp\tuxconfig*

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Service status	Indicates the current state of this service.		This measure reports the value Available if the service is up and running. The numeric value that corresponds to this measure value is 100. Note: By default, this measure reports the **Measure Value** mentioned above to indicate the current state of a server . The graph of this measure however,

Measurement	Description	Measurement Unit	Interpretation
			represents the same using the numeric equivalent only.
Requests handled	Indicates the number of requests handled by this service during the last measurement period.	Number	A very high value could indicate that the service is overloaded with requests. You can compare the value of this measure across services to know which services have the maximum number of requests; this way, you can find out which applications are most popular.

4.2.3 Tuxedo Queues Test

The BEA Tuxedo message queuing servers provide for time-independent communication among clients and servers in a BEA Tuxedo application. They make it possible for an application, within a global transaction, to store client and server generated messages to stable storage for processing later. A client or server process involved in message queuing communications decides when it wants to retrieve a message off its queue.

Since too many enqueued messages can cause performance slowdowns, it is good practice to monitor the length of each message queue, identify overloaded queues, and initiate measures to reduce the load on the queue. The **Tuxedo Queues** test provides these queue-level insights.

Target of the test : A Tuxedo Domain Server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for every *Machine_name:Queue_name* associated with the Tuxedo Domain server being monitored.

Configurable parameters for the test

Parameter	Description
Test period	How often should the test be executed.
Host	The host for which the test is to be configured.
Port	The port at which the specified host listens. The default port is *12345*.
TuxDir	Specify the full path to the installation directory of the Tuxedo Domain server. For

Parameter	Description
	example, while monitoring a Tuxedo server on Windows, your TuxDir can be: *E:\oraclehome\tuxedo11gR1*
AppDir	Specify the full path to the application directory of the Tuxedo Domain server. For example, while monitoring a Tuxedo server on Windows, your appdir can be: *E:\oraclehome\tuxedo11gR1\samples\atmi\simpapp*
TuxConfig	Each BEA Tuxedo domain is controlled by a configuration file in which installation-dependent parameters are defined. The binary version of this configuration file is called TuxConfig. The TUXCONFIG file may be given any name; the actual name is the device or system filename specified in the TUXCONFIG environment variable. Specify the name of the TuxConfig file here. For example, while monitoring a Tuxedo server on Windows, your TuxConfig can be:*E:\oraclehome\tuxedo11gR1\samples\atmi\simpapp\tuxconfig*

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Servers connected	Indicates the number of servers currently connected to this queue.	Number	
Average queue length	Indicates the average number of requests in this queue.	Number	A high value or a consistent increase in this value could indicate a potential overload condition.
Service requests queued	Indicates the number of service requests in this queue.	Number	
All requests in queue	Indicates the actual number of requests in this queue.	Number	

4.2.4 Tuxedo Connections Test

Several types of applications require the notion of a conversation with a server during which context is kept from message to message. Application programmers can use the conversational functions to establish and maintain state- preserving connections between the requesting process and conversational server processes. Specifically, programmers can do the following:

- Open a connection to a conversational server
- Begin and end a global transaction during the conversation
- Have a conversation span multiple machines and resource managers
- Detect and provide notification of connection failures
- Terminate the connection when satisfied that the task has been completed

A conversational server is dedicated to the originating requester for the duration of the connection; the system automatically spawns a new copy of a server if one is not available when a conversational connection is requested.

This test monitors the number of conversational connections that are currently active on the Tuxedo server.

Target of the test : A Tuxedo Domain Server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for every Tuxedo Domain server being monitored.

Configurable parameters for the test

Parameter	Description
Test period	How often should the test be executed.
Host	The host for which the test is to be configured.
Port	The port at which the specified host listens. The default port is *12345*.
TuxDir	Specify the full path to the installation directory of the Tuxedo Domain server. For example, while monitoring a Tuxedo server on Windows, your TuxDir can be: *E:\oraclehome\tuxedo11gR1*
AppDir	Specify the full path to the application directory of the Tuxedo Domain server. For example, while monitoring a Tuxedo server on Windows, your appdir can be: *E:\oraclehome\tuxedo11gR1\samples\atmi\simpapp*
TuxConfig	Each BEA Tuxedo domain is controlled by a configuration file in which installation-dependent parameters are defined. The binary version of this configuration file is called TuxConfig. The TUXCONFIG file may be given any name; the actual name is the device or system filename specified in the TUXCONFIG environment variable. Specify the name of the TuxConfig file here. For example, while monitoring a Tuxedo server on Windows, your TuxConfig can be:*E:\oraclehome\tuxedo11gR1\samples\atmi\simpapp\tuxconfig*

Parameter	Description
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
No of connections	Indicates the number of conversational connections currently open on the Tuxedo server.	Number	Use the detailed diagnosis of this measure to view the complete details of the conversational connections. These details include the following: • Originator_GroupId • Originator_Logical_Machine_Id • Data_transmitted_by_originator • Receiver_GroupId • Receiver_Logical_Machine_Id • Data_transmitted_by_receiver • Service_name

4.2.5 Tuxedo Bulletin Boards Test

The BEA Tuxedo system uses the TUXCONFIG file to set up a bulletin board (BB) on each server machine in a Tuxedo domain. When a Tuxedo server process becomes active, it advertises the names of its services in the bulletin board. Some information in the bulletin board is global and is

replicated on every server machine in the Tuxedo domain (for example, the names and locations of all servers offering a particular service). Other information is local and is visible only on the local bulletin board (for example, the actual number and type of client requests currently waiting on a local server request queue).

The bulletin board provides location and namespace transparency within a Tuxedo domain. Location transparency means that Tuxedo client and server processes do not have to be aware of the location of a resource within the Tuxedo domain. Namespace transparency means that Tuxedo client and server processes can use the same naming conventions (and namespace) to locate any resource in the Tuxedo domain.

Using the **Tuxedo Bulletin Boards** test, you can determine how many servers, services, and interfaces are registered with the BB, and what percentage of these elements are currently in use and not.

Target of the test : A Tuxedo Domain Server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for every Tuxedo Domain server being monitored.

Configurable parameters for the test

Parameter	Description
Test period	How often should the test be executed.
Host	The host for which the test is to be configured.
Port	The port at which the specified host listens. The default port is *12345*.
TuxDir	Specify the full path to the installation directory of the Tuxedo Domain server. For example, while monitoring a Tuxedo server on Windows, your TuxDir can be: *E:\oraclehome\tuxedo11gR1*
AppDir	Specify the full path to the application directory of the Tuxedo Domain server. For example, while monitoring a Tuxedo server on Windows, your appdir can be: *E:\oraclehome\tuxedo11gR1\samples\atmi\simpapp*
TuxConfig	Each BEA Tuxedo domain is controlled by a configuration file in which installation-dependent parameters are defined. The binary version of this configuration file is called TuxConfig. The TUXCONFIG file may be given any name; the actual name is the device or system filename specified in the TUXCONFIG environment variable. Specify the name of the TuxConfig file here. For example, while monitoring a Tuxedo server on Windows, your TuxConfig can be:*E:\oraclehome\tuxedo11gR1\samples\atmi\simpapp\tuxconfig*

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Total servers	Indicates the number of servers registered with the BB.	Number	
Total services	Indicates the number of services advertised on the BB.	Number	
Total interfaces	Indicates the number of network interfaces that are registered with the BB.	Number	
Current servers	Indicates the number of servers currently available in the domain.	Number	
Current services	Indicates the number of services currently available in the domain.	Number	
Current interfaces	Indicates the number of interfaces currently available in the domain.	Number	
Current queues	Indicates the number of queues currently available in the domain.	Number	
Available servers	Indicates the percentage of servers registered with the BB that are currently available.	Percent	Ideally, the value of this measure should be high.
Available services	Indicates the percentage of services advertised on the BB that are currently available.	Percent	Ideally, the value of this measure should be high.
Available interfaces	Indicates the percentage of interfaces registered with the BB tha are currently available.	Percent	Ideally, the value of this measure should be high.

4.2.6 Tuxedo Clients Test

Monitoring the clients that are communicating with the Tuxedo server and the users who login from those clients will accurately point you to the busy users and the transaction load they impose on the server. In addition, it will also shed light on the idle users and the duration of the idle sessions, so that users who have been idle for too long a time can be isolated and their sessions terminated to avoid unnecessary resource usage. The **Tuxedo Clients** test provides such performance insights pertaining to every user who connects to the server from each client.

This test is disabled by default. To enable the test, go to the **ENABLE / DISABLE TESTS** page using the menu sequence : Agents -> Tests -> Enable/Disable, select Tuxedo as the **Component type**, set *Performance* as the **Test type**, choose the test from the **DISABLED TESTS** list, and click on the **<** button to move the test to the **ENABLED TESTS** list. Finally, click the **Update** button.

Target of the test : A Tuxedo Domain Server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for every *username:clientname* connecting to the Tuxedo Domain server being monitored.

Configurable parameters for the test

Parameter	Description
Test period	How often should the test be executed.
Host	The host for which the test is to be configured.
Port	The port at which the specified host listens. The default port is *12345*.
TuxDir	Specify the full path to the installation directory of the Tuxedo Domain server. For example, while monitoring a Tuxedo server on Windows, your TuxDir can be: *E:\oraclehome\tuxedo11gR1*
AppDir	Specify the full path to the application directory of the Tuxedo Domain server. For example, while monitoring a Tuxedo server on Windows, your appdir can be: *E:\oraclehome\tuxedo11gR1\samples\atmi\simpapp*
TuxConfig	Each BEA Tuxedo domain is controlled by a configuration file in which installation-dependent parameters are defined. The binary version of this configuration file is called TuxConfig. The TUXCONFIG file may be given any name; the actual name is the device or system filename specified in the TUXCONFIG environment variable. Specify the name of the TuxConfig file here. For example, while monitoring a Tuxedo server on

Parameter	Description
	Windows, your TuxConfig can be:*E:\oraclehome\tuxedo11gR1\samples\atmi\simpapp\tuxconfig*

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Client status	Indicates the current state of this client.	Number	The values this measure can report are as follows: • idle • busy • transaction idle • transaction busy The numeric values that correspond to the measure values listed above are as follows: **Measure Value** / **Numeric Value** Idle / 0 Transaction Idle / 1 Busy / 90 Transaction Busy / 100 **Note:** By default, this measure reports the **Measure Value**s discussed in the table above to indicate the current state of a client. The graph of this measure however, represents the same using the numeric equivalents only.
Total logged time	Indicates the total time for which this client was logged into the server	Mins	If any client was in the idle state for too long a time on the server, it is a cause for concern.

Measurement	Description	Measurement Unit	Interpretation
Begins	Indicates the number of transactions that have begun.	Number	
Commits	Indicates the number of transactions that were committed.	Number	
Aborts	Indicates the number of transactions that were aborted.	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.