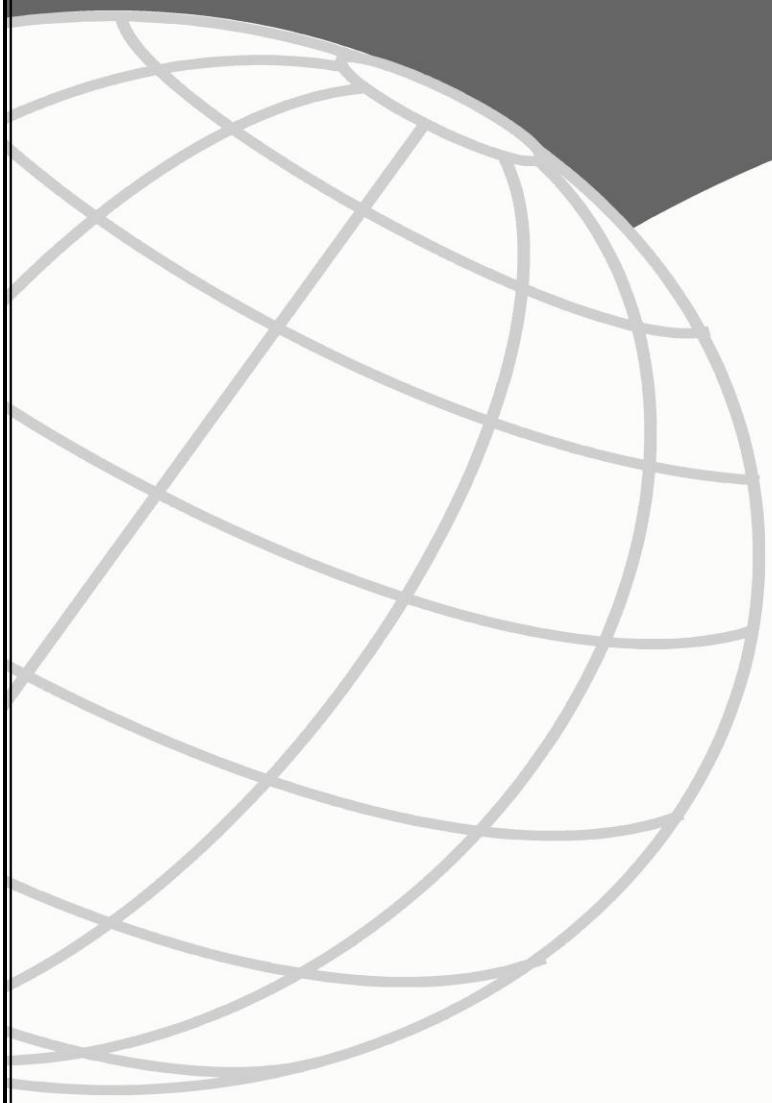

Monitoring the Tuxedo Domain Servers

eG Enterprise v6

Restricted Rights Legend

Trademarks

Copyright

Table of Contents

Table of Figures

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.

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.

This document discusses both these models in detail.

2 Monitoring the Tuxedo Domain Servers Running on Windows/Solaris Systems

Figure 2.1 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.

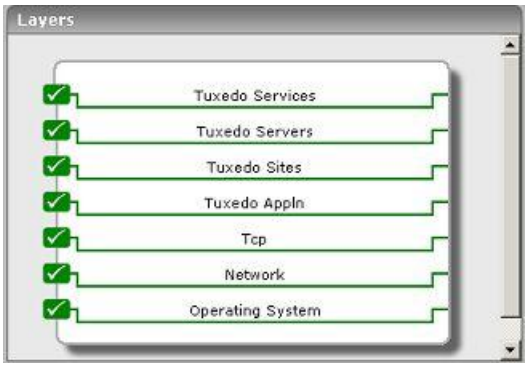

Figure 2.1: Layer model of a Tuxedo Domain server

The sections to come will elaborately discuss each of the top 4 layers of Figure 2.1. For details on the 3 layers at the bottom of Figure 2.1, please refer to the *Monitoring Unix and Windows Servers* document.

2.1 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 2.2: Tests executing on the Tuxedo Appln layer

2.1.1 Tuxedo AR Time Test

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

Purpose	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

<table>
<tr><td>Configurable parameters for the test</td><td colspan="3">1. TEST PERIOD - How often should the test be executed
2. HOST - The host for which the test is to be configured
3. 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.
4. 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.
5. 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.
6. TUXDIR - Specify the installation directory of the Tuxedo domain server.
7. USERNAME - If security is enabled for an application, then provide the username that can be used to connect to the application. Otherwise, enter NONE.
8. PASSWORD - If security is enabled for an application, then provide the password for the above-mentioned Username. Otherwise, enter NONE.
9. CONFIRM PASSWORD – Confirm the password (if specified) by retyping it here.
10. TUXCONFIG - Provide the full path to the tuxconfig file used to run the Tuxedo application.</td></tr>
<tr><td>Outputs of the test</td><td colspan="3">One set of results for the server being monitored</td></tr>
<tr><td>Measurements made by the test</td><td>Measurement</td><td>Measurement Unit</td><td>Interpretation</td></tr>
</table>

	Availability: Indicates the availability of the Tuxedo application.	Percent	
	Response time: Indicates the response time of the Tuxedo application.	Sec	

2.1.2 Tuxedo Domain Test

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

Purpose	Reports 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

Configurable parameters for the test	1. **TEST PERIOD** - How often should the test be executed 2. **HOST** - The host for which the test is to be configured 3. **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. 4. **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. 5. **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.** 6. **TUXDIR** - Specify the installation directory of the Tuxedo domain server. 7. **USERNAME** - If security is enabled for an application, then provide the username that can be used to connect to the application. Otherwise, enter NONE. 8. **PASSWORD** - If security is enabled for an application, then provide the password for the above-mentioned Username. Otherwise, enter NONE. 9. **CONFIRM PASSWORD** – Confirm the password (if specified) by retyping it here. 10. **TUXCONFIG** - Provide the full path to the *tuxconfig* file used to run the Tuxedo application.		
Outputs of the test	One set of results for the server being monitored		
Measurements made by the test	**Measurement**	**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	

2.1.3 Tuxedo TQueues Test

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

Purpose	Reports measurements pertaining to queues in a Tuxedo application
Target of the	A Tuxedo Domain server

<table>
<tr><td>test</td><td colspan="3"></td></tr>
<tr><td>Agent deploying the test</td><td colspan="3">An internal agent</td></tr>
<tr><td>Configurable parameters for the test</td><td colspan="3">1. TEST PERIOD - How often should the test be executed
2. HOST - The host for which the test is to be configured
3. 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.
4. 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.
5. 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.
6. TUXDIR - Specify the installation directory of the Tuxedo domain server.
7. USERNAME - If security is enabled for an application, then provide the username that can be used to connect to the application. Otherwise, enter NONE.
8. PASSWORD - If security is enabled for an application, then provide the password for the above-mentioned Username. Otherwise, enter NONE.
9. CONFIRM PASSWORD – Confirm the password (if specified) by retyping it here.
10. TUXCONFIG - Provide the full path to the tuxconfig file used to run the Tuxedo application.</td></tr>
<tr><td>Outputs of the test</td><td colspan="3">One set of results for the server being monitored</td></tr>
<tr><td>Measurements made by the</td><td>Measurement</td><td>Measurement Unit</td><td>Interpretation</td></tr>
</table>

<table>
<tr><td rowspan="6">test</td><td>Queue state:
Indicates the state of the queue</td><td>Number</td><td colspan="3">It could take any of the following values:</td></tr>
<tr><td></td><td></td><td>Value</td><td>State</td><td>Explanation</td></tr>
<tr><td></td><td></td><td>1</td><td>Active</td><td>Indicates that at least one server associated with the queue object is active</td></tr>
<tr><td></td><td></td><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></td><td></td><td>3</td><td>Suspended</td><td>Indicates that the server(s) associated with the queue object are currently in the suspended state</td></tr>
<tr><td></td><td></td><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>
<tr><td>Server count:
Indicates the number of active servers associated with the queue</td><td>Number</td><td colspan="3"></td></tr>
<tr><td>Request queued:
Indicates the number of requests currently queued in the queue</td><td>Number</td><td colspan="3"></td></tr>
<tr><td></td><td>Current workload:
Indicates the workload currently associated with the queue</td><td>Number</td><td colspan="3"></td></tr>
</table>

2.2 The Tuxedo Sites Layer

Error! Reference source not found. 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 2.3: Tests executing on the Tuxedo Sites layer

2.2.1 Tuxedo Bridges Test

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

Purpose	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

<table>
<tr><td>Configurable parameters for the test</td><td colspan="4">1. TEST PERIOD - How often should the test be executed
2. HOST - The host for which the test is to be configured
3. 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.
4. 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.
5. 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.
6. TUXDIR - Specify the installation directory of the Tuxedo domain server.
7. USERNAME - If security is enabled for an application, then provide the username that can be used to connect to the application. Otherwise, enter NONE.
8. PASSWORD - If security is enabled for an application, then provide the password for the above-mentioned Username. Otherwise, enter NONE.
9. CONFIRM PASSWORD – Confirm the password (if specified) by retyping it here.
10. TUXCONFIG - Provide the full path to the tuxconfig file used to run the Tuxedo application.</td></tr>
<tr><td>Outputs of the test</td><td colspan="4">One set of results for the server being monitored</td></tr>
<tr><td rowspan="2">Measurements made by the test</td><td>Measurement</td><td>Measurement Unit</td><td colspan="3">Interpretation</td></tr>
<tr><td>Bridge state:
Indicates the state of a bridge between two sites in</td><td>Number</td><td colspan="3">It could take any of the following values:
Value | State | Explanation</td></tr>
</table>

	a Tuxedo application		1	Active	Indicates that the bridge or connection between two sites in a Tuxedo application is active or established
			2	Suspended	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
			3	Pending	Indicates that an asynchronous bridge or connection has been requested, but has not yet been completed. The final outcome of the connection request has not been determined.
			4	Inactive	Indicates that the bridge or connection between two sites is inactive
	Data receive rate: Indicates the rate at which bytes are sent from the destination logical machine to the source logical machine	KB/sec			
	Data transmit rate: Indicates the rate at which bytes are sent from the source logical machine to the destination logical machine	KB/sec			

	Message receive rate: Indicates the rate at which messages are sent from the destination logical machine to the source logical machine	Msgs/sec	
	Message transmit rate: Indicates the rate at which messages are sent from the source logical machine to the destination logical machine	Msgs/sec	

2.2.2 Tuxedo Sites Test

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

Purpose	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

Configurable parameters for the test	1. **TEST PERIOD** - How often should the test be executed 2. **HOST** - The host for which the test is to be configured 3. **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. 4. **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. 5. **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.** 6. **TUXDIR** - Specify the installation directory of the Tuxedo domain server. 7. **USERNAME** - If security is enabled for an application, then provide the username that can be used to connect to the application. Otherwise, enter NONE. 8. **PASSWORD** - If security is enabled for an application, then provide the password for the above-mentioned Username. Otherwise, enter NONE. 9. **CONFIRM PASSWORD** – Confirm the password (if specified) by retyping it here. 10. **TUXCONFIG** - Provide the full path to the *tuxconfig* file used to run the Tuxedo application.
Outputs of the test	One set of results for the server being monitored

<table>
<tr><th>Measurements made by the test</th><th>Measurement</th><th>Measurement Unit</th><th colspan="3">Interpretation</th></tr>
<tr><td rowspan="5"></td><td rowspan="4">Site state:
Indicates the state of a site configured under a Tuxedo domain</td><td rowspan="4">Number</td><td colspan="3">It could take any of the following values:</td></tr>
<tr><td>Value</td><td>State</td><td>Explanation</td></tr>
<tr><td>1</td><td>Active</td><td>Indicates that the site is defined and active</td></tr>
<tr><td>2</td><td>Partitioned</td><td>Indicates that the site defined and active, but is currently unreachable. This state is equivalent to Active for the purpose of determining permissions.</td></tr>
<tr><td>3</td><td>Inactive</td><td>Indicates that the site is defined and inactive</td></tr>
<tr><td>Transaction aborts:
Indicates the number of clients, both native and Workstation, who are currently logged in to the site</td><td>Number</td><td colspan="3"></td></tr>
<tr><td>Transaction commits:
Indicates the number of Workstation clients, currently logged in to the site</td><td>Number</td><td colspan="3"></td></tr>
<tr><td>TListen state:
Indicates the state of the Tlisten process running on this site</td><td>Number</td><td colspan="3">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></td><td>Total clients:
Indicates the number of clients, both native and Workstation, who are currently logged in to the site</td><td>Number</td><td colspan="3"></td></tr>
</table>

	Workstation clients: Indicates the number of Workstation clients, currently logged in to the site	Number	
	Conversations: Indicates the number of active conversations with participants on the site	Number	
	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	

2.3 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 2.4: Tests associated with the Tuxedo Servers layer

2.3.1 Tuxedo Server Test

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

Purpose	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

<table>
<tr><td>Configurable parameters for the test</td><td colspan="4">1. TEST PERIOD - How often should the test be executed
2. HOST - The host for which the test is to be configured
3. 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.
4. 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.
5. 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.
6. TUXDIR - Specify the installation directory of the Tuxedo domain server.
7. USERNAME - If security is enabled for an application, then provide the username that can be used to connect to the application. Otherwise, enter NONE.
8. PASSWORD - If security is enabled for an application, then provide the password for the above-mentioned Username. Otherwise, enter NONE.
9. CONFIRM PASSWORD – Confirm the password (if specified) by retyping it here.
10. TUXCONFIG - Provide the full path to the tuxconfig file used to run the Tuxedo application.</td></tr>
<tr><td>Outputs of the test</td><td colspan="4">One set of results for the server being monitored</td></tr>
<tr><td rowspan="2">Measurements made by the test</td><td>Measurement</td><td>Measurement Unit</td><td colspan="3">Interpretation</td></tr>
<tr><td>Server state:
Indicates the state of a server running on a</td><td>Number</td><td colspan="3">It could take any of the following values:
Value | State | Explanation</td></tr>
</table>

	particular site		1	Active	Indicates that the server object is defined and active
			2	Migrating	Indicates that the server is defined and currently in a state of migration to the server group's secondary logical machine
			3	Cleaning	Indicates that the server object is defined and is currently being cleaned up by the system after an abnormal death
			4	Restarting	Indicates that the server object defined and is currently being restarted by the system after an abnormal death
			5	Suspended	Indicates that the server object is defined and is currently suspended pending shutdown
			6	Partitioned	Indicates that the server object is defined and active; however, the machine where the server is running is currently partitioned from the master site
			7	Dead	Indicates that the server object is defined, identified as active in the bulletin board, but currently not running due to an abnormal death
			8	Inactive	Indicates that the server object is defined and inactive

	Workload completions: Indicates the rate at which workload was completed by the server	Works/Sec	
	Total conversations: Indicates the total number of conversations initiated by the server	Number	
	Active conversations: Indicates the number of currently active conversations initiated by the server	Number	
	Total requests: Indicates the total number of requests made by the server via tpcall() or tpacall()	Number	
	Active requests: Indicates the number of currently active requests made by the server via tpcall() or tpacall()	Number	
	Transaction initiations: Indicates the rate at which transactions were begun by the server since it was last restarted	Trans/sec	
	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	
	Request completions: Indicates the rate at which requests are completed by the server	Reqs/Sec	

2.3.2 Tuxedo Processes Test

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

Purpose	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
Configurable parameters for the test	1. **TEST PERIOD** - How often should the test be executed 2. **HOST** - The host for which the test is to be configured 3. **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. 4. **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. 5. **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**. 6. **TUXDIR** - Specify the installation directory of the Tuxedo domain server. 7. **USERNAME** - If security is enabled for an application, then provide the username that can be used to connect to the application. Otherwise, enter NONE. 8. **PASSWORD** - If security is enabled for an application, then provide the password for the above-mentioned Username. Otherwise, enter NONE. 9. **CONFIRM PASSWORD** – Confirm the password (if specified) by retyping it here. 10. **TUXCONFIG** - Provide the full path to the *tuxconfig* file used to run the Tuxedo application.

Outputs of the test	One set of results for the server being monitored		
Measurements made by the test	**Measurement**	**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	

2.4 The Tuxedo Services Layer

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

Figure 2.5: Tests that execute on the Tuxedo Services layer

2.4.1 Tuxedo Service Test

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

Purpose	Measures the performance of services provided by servers running on a Tuxedo application

<table>
<tr><td>Target of the test</td><td colspan="3">A Tuxedo Domain server</td></tr>
<tr><td>Agent deploying the test</td><td colspan="3">An internal agent</td></tr>
<tr><td>Configurable parameters for the test</td><td colspan="3">1. TEST PERIOD - How often should the test be executed
2. HOST - The host for which the test is to be configured
3. 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.
4. 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.
5. 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.
6. TUXDIR - Specify the installation directory of the Tuxedo domain server.
7. USERNAME - If security is enabled for an application, then provide the username that can be used to connect to the application. Otherwise, enter NONE.
8. PASSWORD - If security is enabled for an application, then provide the password for the above-mentioned Username. Otherwise, enter NONE.
9. CONFIRM PASSWORD – Confirm the password (if specified) by retyping it here.
10. TUXCONFIG - Provide the full path to the tuxconfig file used to run the Tuxedo application.</td></tr>
<tr><td>Outputs of the test</td><td colspan="3">One set of results for the server being monitored</td></tr>
<tr><td>Measurements made by the</td><td>Measurement</td><td>Measurement Unit</td><td>Interpretation</td></tr>
</table>

<table>
<tr><td rowspan="6">test</td><td>Service state:
Indicates the state of a service provided by a server in a Tuxedo domain.</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 service object is defined and active</td></tr><tr><td>2</td><td>Suspended</td><td>Indicates that the service object is defined and is currently suspended. The service is not available for access by the application in this state</td></tr><tr><td>3</td><td>Partitioned</td><td>Indicates that the service object is defined and active, but is currently partitioned from the master site</td></tr><tr><td>4</td><td>Inactive</td><td>Indicates that the service object is defined and inactive</td></tr></table></td></tr>
<tr><td>Load imposed:
Indicates the load imposed by the service object on the system</td><td>Number</td><td>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.</td></tr>
<tr><td>Requests completed:
Indicates the rate at which requests are completed by the service object</td><td>Reqs/Sec</td><td></td></tr>
<tr><td>Requests queued:
Indicates the number of requests currently enqueued to the service</td><td>Number</td><td></td></tr>
</table>

3

Platform-Independent Monitoring of Tuxedo Domain Servers

Figure 3.1 depicts the *Tuxedo* monitoring model, which is capable of extracting performance metrics from Tuxedo Domain servers executing on any Windows or Unix platform.

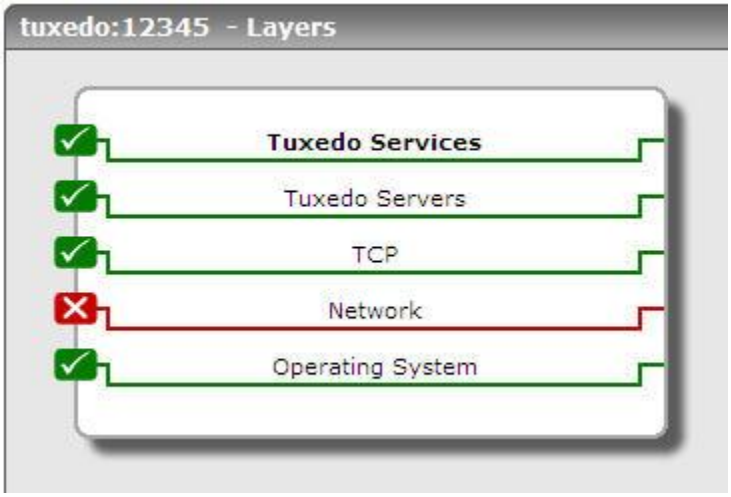

Figure 3.1: The Tuxedo monitoring model

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 Figure 3.1 have already been dealt with the *Monitoring Unix and Windows Servers* document, the sections that follow wil discuss the top two layers only.

3.1 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 3.2: The tests mapped to the Tuxedo Servers layer

3.1.1 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.

Purpose	Periodically monitor the usage of each connection pool and promptly isolates pools with insufficient connections
Target of the test	A Tuxedo Domain server
Agent deploying the test	An internal agent

Configurable parameters for the test	1. **TEST PERIOD** - How often should the test be executed 2. **HOST** - The host for which the test is to be configured 3. **PORT** - The port at which the specified host listens. The default port is 12345. 4. **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* 5. **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* 6. **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 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*		
Outputs of the test	One set of results for every *Group_name:Pool_name* auto-discovered from the Tuxedo server being monitored		
Measurements made by the test	**Measurement**	**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.

3.1.2 Tuxedo Domain Server Test

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

a. **BRIDGE**: An administration server that establishes the machine's listening address.

b. **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.

Purpose	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
Configurable parameters for the test	1. **TEST PERIOD** - How often should the test be executed 2. **HOST** - The host for which the test is to be configured 3. **PORT** - The port at which the specified host listens. The default port is 12345. 4. **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* 5. **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* 6. **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 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*
Outputs of the test	One set of results for every *Servername:ServerID* associated with each administration server on the Tuxedo Domain server being monitored

<table>
<tr><th>Measurements made by the test</th><th>Measurement</th><th>Measurement Unit</th><th>Interpretation</th></tr>
<tr><td></td><td>Current server state:
Indicates the current state of this administration server.</td><td></td><td>The values that this measure can report and their corresponding numeric equivalents have been detailed in the table below:
<table><tr><th>Measure Value</th><th>Numeric Value</th></tr><tr><td>Active</td><td>100</td></tr><tr><td>Idle</td><td>1</td></tr><tr><td>Inactive</td><td>0</td></tr></table>
Note:
By default, this measure reports the Measure Values 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.</td></tr>
<tr><td></td><td>Requests handled:
Indicates the number of requests handled by this administration server during the last measurement period.</td><td>Number</td><td></td></tr>
<tr><td></td><td>Server load:
Indicates the load on this server during the last measurement period.</td><td>Number</td><td>Compare the value of this measure across servers to identify the server that is overloaded.</td></tr>
</table>

3.1.3 Tuxedo application server load Test

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

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, pick *Tuxedo* as the **Component type**, *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.

This test is applicable for Tuxedo Domain servers of version 6.5 and above.

<table>
<tr><td>Purpose</td><td colspan="3">Reports the current availability of and the load on every server on the target Tuxedo server, so that server failures and improper load distribution amongst servers (i.e., services) are detected early and fixed</td></tr>
<tr><td>Target of the test</td><td colspan="3">A Tuxedo Domain server</td></tr>
<tr><td>Agent deploying the test</td><td colspan="3">An internal agent</td></tr>
<tr><td>Configurable parameters for the test</td><td colspan="3">1. TEST PERIOD - How often should the test be executed
2. HOST - The host for which the test is to be configured
3. PORT - The port at which the specified host listens. The default port is 12345.
4. APPNAME - Specify the name of an application running on the Tuxedo server. This test will report metrics for every server related to this application.</td></tr>
<tr><td>Outputs of the test</td><td colspan="3">One set of results for every server on the Tuxedo Domain server being monitored</td></tr>
<tr><td rowspan="4">Measurements made by the test</td><td>Measurement</td><td>Measurement Unit</td><td>Interpretation</td></tr>
<tr><td>Availability:
Indicates the current availability of this server.</td><td>Percent</td><td>If the value of this measure is 100, then it indicates that the server is available. A value of 0 indicates that the server is currently unavailable.</td></tr>
<tr><td>Request handled rate:
Indicates the rate at which the requests were handled by this server during the last measurement period.</td><td>Req/sec</td><td>This measure is a clear indicator of the load on the servers of the Tuxedo Application.
Comparing the value of this measure across servers will help you to identify the server that is busy handling the maximum number of requests.</td></tr>
<tr><td>Load done:
Indicates the load on this server during the last measurement period.</td><td>Number</td><td>A higher value indicates that the server is currently overloaded.
Compare the value of this measure across servers to identify the server that is overloaded.</td></tr>
</table>

3.1.4 Tuxedo queue activities 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 queue activities** test provides these queue-level insights.

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, pick *Tuxedo* as the **Component type**, *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.

This test is applicable for Tuxedo Domain servers of version 6.5 and above.

Purpose	Monitors the length of each message queue, identifies overloaded queues, and initiates measures to reduce the load on the queue		
Target of the test	A Tuxedo Domain server		
Agent deploying the test	An internal agent		
Configurable parameters for the test	1. **TEST PERIOD** - How often should the test be executed 2. **HOST** - The host for which the test is to be configured 3. **PORT** - The port at which the specified host listens. The default port is 12345. 4. **APPNAME** - Specify the name of an application running on the Tuxedo server. This test will report metrics for every queue related to the server of this application.		
Outputs of the test	One set of results for every *application:servername:queuename* connecting to the Tuxedo Domain server being monitored		
Measurements made by the test	**Measurement**	**Measurement Unit**	**Interpretation**
	Servers in queue: Indicates the number of servers that are currently connected to this queue.	Number	

	Queued requests: Indicates the number of requests that are currently in this queue.	Number	A low value is desired for this measure. A high value or a consistent incease in this value is indicative of server overload condition which in turn leads to the poor performance of the server. Compare the value of this measure across the queues to identify the queue that is stacking the maximum number of requests thus overloading the servers.

3.1.5 Tuxedo client processes 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 client processes** 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, pick *Tuxedo* as the **Component type**, *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.

This test is applicable for Tuxedo Domain servers of version 6.5 and above.

Purpose	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
Target of the test	A Tuxedo Domain server
Agent deploying the test	An internal agent

<table>
<tr><td>Configurable parameters for the test</td><td colspan="3">1. TEST PERIOD - How often should the test be executed
2. HOST - The host for which the test is to be configured
3. PORT - The port at which the specified host listens. The default port is 12345.
4. APPNAME - Specify the name of an application running on the Tuxedo server. This test will report metrics for every service related to this application.
5. DETAILED DIAGNOSIS - To make diagnosis more efficient and accurate, the eG system 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 against DETAILED DIAGNOSIS. 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:
o The eG manager license should allow the detailed diagnosis capability
o Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.</td></tr>
<tr><td>Outputs of the test</td><td colspan="3">One set of results for each client of the Tuxedo application that is to be monitored</td></tr>
<tr><td rowspan="2">Measurements made by the test</td><td>Measurement</td><td>Measurement Unit</td><td>Interpretation</td></tr>
<tr><td>Number of client requests:
Indicates the number of requests received from this client.</td><td>Number</td><td>The detailed diagnosis of this test if enabled lists the name of the machine, user who is connected to the Tuxedo server, the name of the client, TIME, the status of the client(for eg., IDLE, IDLET, BUSY, BUSYT), the transaction status (Begin, Abort, Commit).</td></tr>
</table>

3.2 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 3.3: The tests mapped to the Tuxedo Services layer

3.2.1 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.

Purpose	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

<table>
<tr><td>Configurable parameters for the test</td><td colspan="3">1. TEST PERIOD - How often should the test be executed
2. HOST - The host for which the test is to be configured
3. PORT - The port at which the specified host listens. The default port is 12345.
4. 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
5. 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
6. 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 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
7. DETAILED DIAGNOSIS - To make diagnosis more efficient and accurate, the eG system 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 against DETAILED DIAGNOSIS. 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:
o The eG manager license should allow the detailed diagnosis capability
o Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.</td></tr>
<tr><td>Outputs of the test</td><td colspan="3">One set of results for the Tuxedo Domain server being monitored</td></tr>
<tr><td>Measurements made by the</td><td>Measurement</td><td>Measurement Unit</td><td>Interpretation</td></tr>
</table>

test	**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: a. TMGACTIVE b. TMGABORTED c. TMGTOBEABORTED d. TMGCOMMITED e. TMGCOMMITCALLED f. TMGDECIDED

3.2.2 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.

Purpose	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

Configurable parameters for the test	1. TEST PERIOD - How often should the test be executed 2. Host - The host for which the test is to be configured 3. port - The port at which the specified host listens. The default port is 12345. 4. 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* 5. 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* 6. 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 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*		
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		
Measurements made by the test	**Measurement**	**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, 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.

3.2.3 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.

Purpose	Monitors the length of each message queue, identifies overloaded queues, and initiates measures to reduce the load on the queue
Target of the test	A Tuxedo Domain server
Agent deploying the test	An internal agent
Configurable parameters for the test	1. **TEST PERIOD** - How often should the test be executed 2. **HOST** - The host for which the test is to be configured 3. **PORT** - The port at which the specified host listens. The default port is 12345. 4. **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* 5. **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* 6. **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 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*
Outputs of the test	One set of results for every *Machine_name:Queue_name* associated with the Tuxedo Domain server being monitored

Measurements made by the test	Measurement	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	

3.2.4 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.

Purpose	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

<table>
<tr><td>Configurable parameters for the test</td><td colspan="3">1. TEST PERIOD - How often should the test be executed
2. HOST - The host for which the test is to be configured
3. PORT - The port at which the specified host listens. The default port is 12345.
4. 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
5. 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
6. 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 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
7. DETAILED DIAGNOSIS - To make diagnosis more efficient and accurate, the eG system 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 against DETAILED DIAGNOSIS. To disable the capability, click on the Off option.
The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled:
○ The eG manager license should allow the detailed diagnosis capability
○ Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.</td></tr>
<tr><td>Outputs of the test</td><td colspan="3">One set of results for every Tuxedo Domain server being monitored</td></tr>
<tr><td rowspan="2">Measurements made by the test</td><td>Measurement</td><td>Measurement Unit</td><td>Interpretation</td></tr>
<tr><td>No of connections:
Indicates the number of conversational connections currently open on the Tuxedo server.</td><td>Number</td><td>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</td></tr>
</table>

3.2.5 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.

Purpose	Helps 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		
Configurable parameters for the test	1. **TEST PERIOD** - How often should the test be executed 2. **HOST** - The host for which the test is to be configured 3. **PORT** - The port at which the specified host listens. The default port is 12345. 4. **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* 5. **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* 6. **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 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*		
Outputs of the test	One set of results for every Tuxedo Domain server being monitored		
Measurements made by the	**Measurement**	**Measurement Unit**	**Interpretation**

test	**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.

3.2.6 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, pick *Tuxedo* as the **Component type**, *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.

Purpose	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
Target of the test	A Tuxedo Domain server
Agent deploying the test	An internal agent
Configurable parameters for the test	1. **TEST PERIOD** - How often should the test be executed 2. **HOST** - The host for which the test is to be configured 3. **PORT** - The port at which the specified host listens. The default port is 12345. 4. **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* 5. **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* 6. **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 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*

<table>
<tr><td>Outputs of the test</td><td colspan="3">One set of results for every username:clientname connecting to the Tuxedo Domain server being monitored</td></tr>
<tr><td rowspan="5">Measurements made by the test</td><td>Measurement</td><td>Measurement Unit</td><td>Interpretation</td></tr>
<tr><td>Client status:
Indicates the current state of this client.</td><td>Number</td><td>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:<table><tr><th>Measure Value</th><th>Numeric Value</th></tr><tr><td>Idle</td><td>0</td></tr><tr><td>Transaction Idle</td><td>1</td></tr><tr><td>Busy</td><td>90</td></tr><tr><td>Transaction Busy</td><td>100</td></tr></table>Note:
By default, this measure reports the Measure Values 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.</td></tr>
<tr><td>Total logged time:
Indicates the total time for which this client was logged into the server</td><td>Mins</td><td>If any client was in the IDLE state for too long a time on the server, it is a cause for concern.</td></tr>
<tr><td>Begins:
Indicates the number of transactions that have begun.</td><td>Number</td><td></td></tr>
<tr><td>Commits:
Indicates the number of transactions that were committed.</td><td>Number</td><td></td></tr>
<tr><td>Aborts:
Indicates the number of transactions that were aborted.</td><td>Number</td><td></td></tr>
</table>

3.2.7 Tuxedo service requests 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 the **Tuxedo service requests** test, you can continuously track the availability 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.

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, pick *Tuxedo* as the **Component type**, *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.

This test is applicable for Tuxedo Domain servers of version 6.5 and above.

Purpose	Helps you to continuously track the availability 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		
Configurable parameters for the test	1. **TEST PERIOD** - How often should the test be executed 2. **HOST** - The host for which the test is to be configured 3. **PORT** - The port at which the specified host listens. The default port is 12345. 4. **APPNAME** - Specify the name of an application running on the Tuxedo server. This test will report metrics for every service related to this application.		
Outputs of the test	One set of results for every *service* associated with each server stored by the application servers on the Tuxedo Domain server being monitored		
Measurements made by the test	**Measurement**	**Measurement Unit**	**Interpretation**
	Availability: Indicates the current availability of this service.	Percent	If the value of this measure is 100, then it indicates that the server is available. A value of 0 indicates that the service is currently unavailable.

	Request handled rate: Indicates the rate at which the requests were handled by this service during the last measurement period.	Req/sec	This measure is a clear indicator of the load on the services of the server. Comparing the value of this measure across services will help you to identify the service that is busy handling the maximum number of requests. This way, you can figure out the applications that are most popular.

4

Conclusion

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

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