



Monitoring Oracle Tuxedo PeopleSoft Internet Architecture (PIA)

eG Innovations Product Documentation

Table of Contents

CHAPTER 1: INTRODUCTION	1
CHAPTER 2: HOW DOES EG ENTERPRISE MONITOR THE TUXEDO PIA SERVER?	4
2.1 Managing the Tuxedo PIA Server	4
CHAPTER 3: MONITORING THE TUXEDO PIA	7
3.1 The Application Processes Layer	8
3.1.1 Tuxedo PIA Application Logs Test	9
3.1.2 PeopleSoft Tuxedo Log Monitor Test	15
3.2 The Tuxedo PIA Server Layer	20
3.2.1 Tuxedo PIA Server Load Test	22
3.2.2 Tuxedo PIA Client Connections Test	25
3.2.3 Tuxedo PIA Server Details Test	28
3.2.4 Tuxedo PIA Database Connections Test	30
3.3 The Tuxedo PIA Service Layer	32
3.3.1 Tuxedo PIA Bulletin Boards Test	33
3.3.2 Tuxedo PIA Server Connections Test	37
3.3.3 Tuxedo PIA Server Queues Test	40
3.3.4 Tuxedo PIA Services Test	42
3.3.5 Tuxedo PIA Server Transactions Test	44
ABOUT EG INNOVATIONS	47

Table of Figures

Figure 1.1: How the PeopleSoft Internet Architecture operates	2
Figure 1.2: How an application domain processes requests	3
Figure 2.1: Adding a Tuxedo PIA Server	5
Figure 2.2: When you attempt to sign out, a list of unconfigured tests will appear as shown in Figure 1.3.	5
Figure 2.3: List of Unconfigured tests to be configured for the Tuxedo PIA Server	5
Figure 2.4: Configuring the Tuxedo PIA Bulletin Board test	6
Figure 3.1: The layer model of a Tuxedo PIA	7
Figure 3.2: The tests mapped to the Application Processes layer	9
Figure 3.3: The detailed diagnosis of the Tuxedo PIA Application Logs Test	15
Figure 3.4: The detailed diagnosis of the PeopleSoft Tuxedo Log Monitor Test	20
Figure 3.5: The tests mapped to the Tuxedo PIA Server layer	22
Figure 3.6: Detailed diagnosis of the Tuxedo PIA Client Connections test	28
Figure 3.7: The detailed diagnosis of the Idle servers measure of the Tuxedo PIA Server Details test	30
Figure 3.8: The tests mapped to the Tuxedo PIA Service layer	33

Chapter 1: Introduction

PeopleSoft Internet Architecture (PIA), leverages a number of internet technologies and concepts to deliver simple, ubiquitous access to PeopleSoft applications and enable the open flow of information between systems. Using PeopleSoft Internet Architecture as the foundation, customers will be able to provide a wide range of end users with access to PeopleSoft applications over the web, as well as more easily integrate their PeopleSoft applications with existing internal systems and external trading partner systems.

The PeopleSoft Internet Architecture is comprised of these main server types:

- RDBMS
- Tuxedo Application Server(s)
- Web server(s)

The servers facilitate connections and process requests from:

- PeopleTools Development Environment: A Windows workstation running a development tool, such as PeopleSoft Application Designer.
- Browser: A supported browser type and version displaying a PeopleSoft application or administrative interface.
- Remote system: A PeopleSoft or third- party system integrated through PeopleSoft Integration Broker's service oriented architecture (SOA).

The following diagram illustrates the relationship between the elements of the PeopleSoft Internet Architecture.

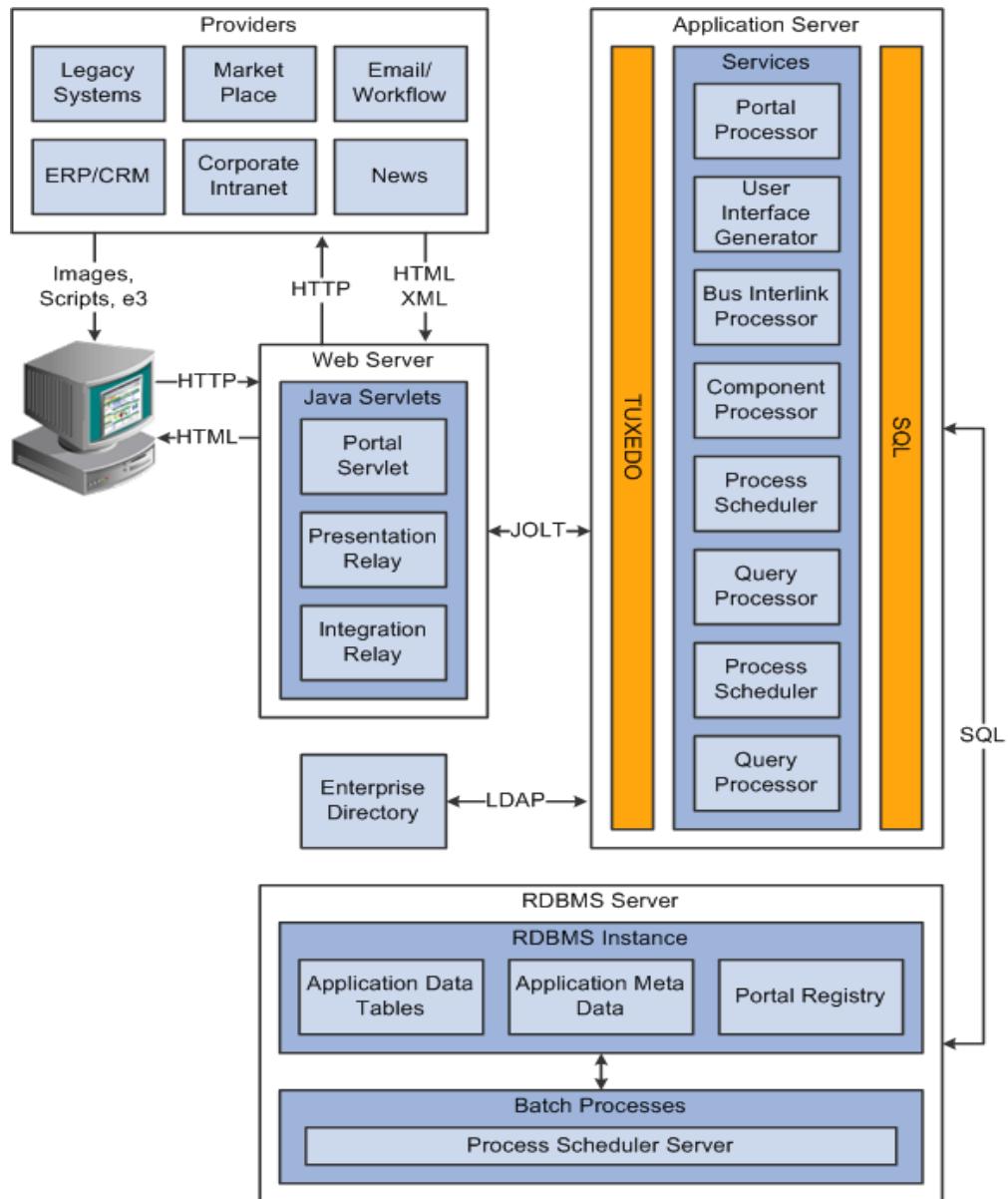


Figure 1.1: How the PeopleSoft Internet Architecture operates

As is evident from the image, application requests are received by the web server using three PeopleSoft servlets – namely, the Presentation Relay Servlet, the Integration Relay Servlet, and the Portal Servlet. When a PeopleSoft application sends a request, it sends a service name and a set of parameters. Any request so received is then forwarded by the web server to the Tuxedo application server. The application server then queues the transaction request to a specific server process that is designed to handle certain services.

An application server consists of numerous PeopleSoft server processes, grouped in domains. When you boot an application server domain, it starts the set of server processes associated with that domain. Each

server process establishes a persistent connection to a PeopleSoft database, and this connection acts as a generic SQL pipeline that the server process uses to send and receive data. Listeners, handlers, and queues in a domain receive requests, route requests, store requests, monitor requests, and return request responses.

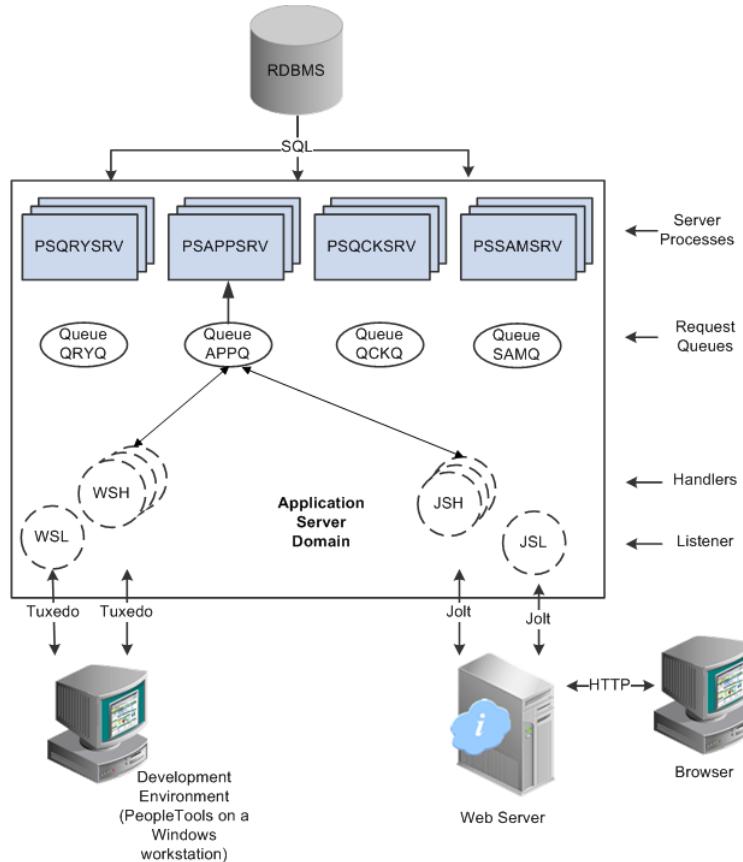


Figure 1.2: How an application domain processes requests

The proper functioning of all these crucial elements – be it the server processes, services, listeners, handlers, and queues in a domain – is essential to ensure that the domain effectively handles application requests, and that users have an above-par experience with PeopleSoft applications at all times.

Chapter 2: How does eG Enterprise Monitor the Tuxedo PIA Server?

eG Enterprise monitors the Tuxedo PIA server only in an agent based manner. The eG agent runs number of tests that native PeopleSoft Server Administration (PSADMIN) commands on a single application domain in the target application server and pull out metrics.

To run PSADMIN commands, each of these tests should be configured with the following:

- The full path to the directory in which the **psadmin.exe** resides;
- The name of the application domain to be monitored;

Note:

For every application domain that you want monitored, you will have to manage a separate *Tuxedo PIA* component in eG.

2.1 Managing the Tuxedo PIA Server

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

1. Log into the eG administrative interface.
2. Follow the Components -> Add/Modify menu sequence in the **Infrastructure** tile of the **Admin** menu.
3. In the **COMPONENT** page that appears next, select Tuxedo PIA as the **Component type**. Then, click the **Add New Component** button. This will invoke Figure 2.1.

COMPONENT

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

Category	Component type
All	Tuxedo PIA

Component information

Host IP/Name	192.168.10.1
Nick name	tuxpia
Port number	9000

Monitoring approach

Agentless	<input type="checkbox"/>
Internal agent assignment	<input checked="" type="radio"/> Auto <input type="radio"/> Manual
External agents	192.168.9.104

Add

Figure 2.1: Adding a Tuxedo PIA Server

4. Specify the **Host IP** and the **Nick name** of the Tuxedo PIA Server in Figure 2.1. Then, click the **Add** button to register the changes.
5. Figure 2.2: When you attempt to sign out, a list of unconfigured tests will appear as shown in Figure 2.3.

List of unconfigured tests for 'Tuxedo PIA'

Performance			tuxpia:9000
PeopleSoft Tuxedo Log Monitor	Tuxedo PIA Application Logs	Tuxedo PIA Bulletin Boards	
Tuxedo PIA Client Connections	Tuxedo PIA Database Connections	Tuxedo PIA Server Connections	
Tuxedo PIA Server Details	Tuxedo PIA Server Load	Tuxedo PIA Server Queues	
Tuxedo PIA Server Transactions	Tuxedo PIA Services		

Figure 2.3: List of Unconfigured tests to be configured for the Tuxedo PIA Server

6. Click on any test in the list of unconfigured tests. For instance, click on the **Tuxedo PIA Bulletin Board** test to configure it. In the page that appears, specify the parameters as shown in Figure 2.4.

Tuxedo PIA Bulletin Boards parameters to be configured for tuxpia:9000 (Tuxedo PIA)

TEST PERIOD	<input type="text" value="5 mins"/>
HOST	<input type="text" value="192.168.10.1"/>
PORT	<input type="text" value="9000"/>
* PSADMIN HOME	<input type="text" value="C:\ps\appsrv and on Unix"/>
* DOMAIN	<input type="text" value="mas"/>

Update

Figure 2.4: Configuring the Tuxedo PIA Bulletin Board test

7. Next, try to sign out of the eG administrative interface, this time you will be prompted to configure **PeopleSoft Tuxedo Log monitor** test and **Tuxedo PIA Applications Logs** test. Configure these tests one after another. To know how to configure the tests, refer to [Monitoring the Tuxedo PIA](#).
8. Finally, signout of the eG administrative interface.

Chapter 3: Monitoring the Tuxedo PIA

eG Enterprise provides a specialized Tuxedo PIA model (see Figure 1.2) for monitoring the Tuxedo Application Server that is at the ‘heart’ of the PeopleSoft Internet Architecture. Using this model, processing bottlenecks within an application domain can be rapidly detected and the reason for these bottlenecks, pinpointed.

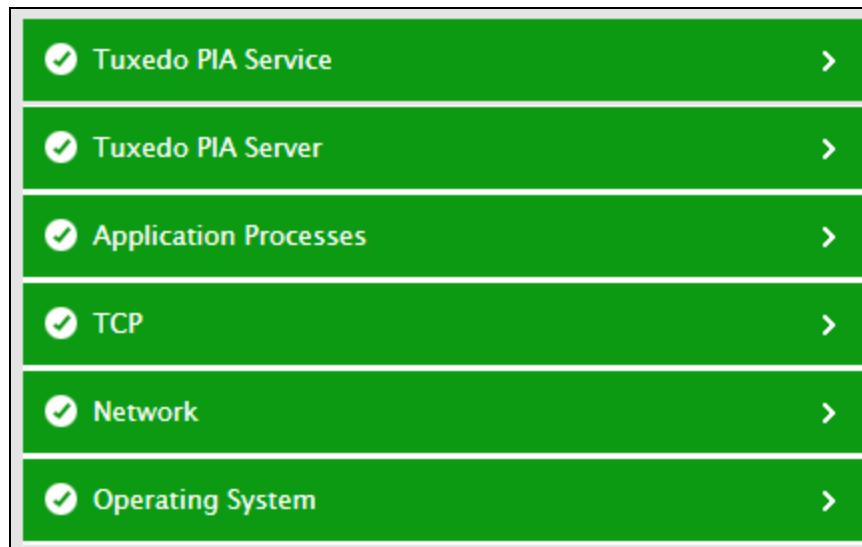


Figure 3.1: The layer model of a Tuxedo PIA

Each layer of Figure 2.1 is mapped to tests that run native PeopleSoft Server Administration (PSADMIN) commands on a single application domain in the target application server and pull out metrics. To run these commands, the pre-requisites discussed in the [How does eG Enterprise Monitor the Tuxedo PIA Server?](#) chapter should be fulfilled.

Once the tests run and collect the desired metrics, administrators can use these domain-specific metrics to find quick and accurate answers to the following persistent performance queries:

- What is the transaction load on the monitored domain?
- Have any transactions to the domain aborted? If so, which ones are these?
- Is any transaction to the domain about to abort? Which is this?
- What is the current workload of the domain? Which client is imposing the maximum load on the domain?

- Has the domain been configured with adequate server process instances, services, and interfaces to handle the load?
- Are all instances of a server process busy servicing requests presently? Are there idle server process instances in the domain?
- Which is the busiest service in the domain? How quickly is this service processing the requests it receives?
- Is any service unavailable?
- Have too many pending requests been enqueued in any request queue in the domain? Which queue is this and which server process is using this queue? How are the requests in this queue impacting the overall workload of the domain? Should more instances of this server process be configured to reduce the queue length, and consequently, the workload?
- Is the Tuxedo application server able to connect to the database? Are there sufficient number of free connections in the database connection pool to facilitate the connection?
- What is the current workload of the conversational server process (PSSAMSrv) in the monitored domain? Which are the conversations that are contributing to this workload? Has the PSSAMSrv server process been configured right to handle this load?
- Have any errors/warnings been captured by the **APPSRV** log files and **TUXLOG** files?

The sections that follow discuss the top 2 layers of the layer model as the rest of the layers have already been discussed in the *Monitoring Unix and Windows Servers* document.

3.1 The Application Processes Layer

Besides monitoring the PeopleSoft-related processes running on the application server host, the tests mapped to this layer also check the Application Server log and Tuxedo log files for error events.

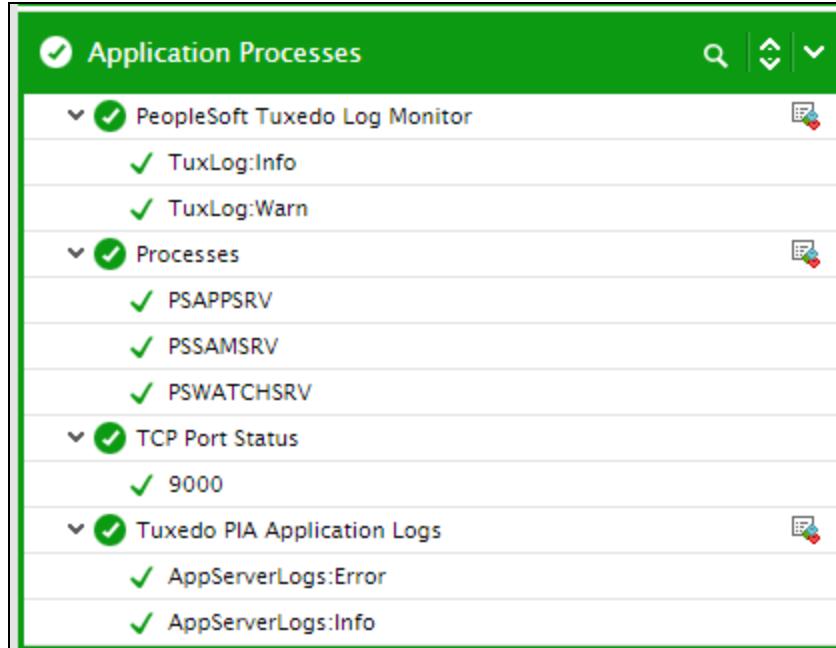


Figure 3.2: The tests mapped to the Application Processes layer

3.1.1 Tuxedo PIA Application Logs Test

The **APPSRV.log** file contains PeopleTools specific logging information. Certificate authentication logs, including information about mismatched distinguished names and certificates that are not in the database, is contained within this log file. In addition, the log file also shows the current service count for an AppServer process, and when a specific process was recycled/spawned. By periodically scanning these log files for specific patterns of log entries, administrators can quickly capture certificate errors, know when a process was spawned, and determine which services are offered by which server process. This is exactly what the **Tuxedo PIA Application Logs** test does. This test periodically searches the `appsr.log` file for specific patterns of error/general information messages and reports the count and nature of such messages, so as to enable administrators to troubleshoot better and to easily obtain configuration details.

Target of the test : A Tuxedo Application Server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for every AlertFile and SearchPattern pair configured.

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The host for which the test is being configured.
Port	Specify the port at which the specified host listens in the Port text box. By default, this is <i>NULL</i> .
AlertFile	<p>This test monitors the APPSRV.log files for errors and configuration information. Typically, these log files will be available in the <PSHome>\appserv\<domain>\LOGS directory and will be named in the following format: <i>APPSRV_<mmdd>.log</i>, where <mmdd> refers to the month and date of creation of the log file. In the LOGS directory therefore, you will find multiple APPSRV.log files, one for every day of the month. To enable the test to monitor a specific APPSRV.log file, you need to provide the full path to that log file against AlertFile. For instance, if <PSHome> (on Windows) is C:\ps, the name of the PeopleSoft domain is psdomain, and the log file you want to monitor was created on the 16th of April, your AlertFile specification will be: C:\ps\appserv\psdomain\LOGS\APPSRV_0416.log. Multiple log files can also be monitored as a comma-separated list. For instance, your AlertFile specification can be: C:\ps\appserv\psdomain\LOGS\APPSRV_0416.log,C:\ps\appserv\psdomain\LOGS\APPSRV_0420.log. Specific log file name patterns can also be specified. For example, to monitor all the log files created in the months of March and April (i.e., in the 3rd and 4th months), the parameter specification can be, : C:\ps\appserv\psdomain\LOGS\APPSRV_03*.log, C:\ps\appserv\psdomain\LOGS\APPSRV_04*.log. Here, '*' indicates leading/trailing characters (as the case may be).</p> <p>Your AlertFile specification can also be of the following format: <i>Name@filepath_or_pattern</i>. Here, <i>Name</i> represents the display name of the path being configured. For instance, to monitor all APPSRV log files that were created in the months of March and April, your specification can be: Marchlogs@ C:\ps\appserv\psdomain\LOGS\APPSRV_03*.log, Aprillogs@C:\ps\appserv\psdomain\LOGS\APPSRV_04*.log. In this case, the display names 'Marchlogs' and 'Aprillogs' will alone be displayed as descriptors of this test.</p> <p>Every time this test is executed, the eG agent verifies the following:</p> <ul style="list-style-type: none"> • Whether any changes have occurred in the size and/or timestamp of the log files that were monitored during the last measurement period; • Whether any new log files (that match the AlertFile specification) have been

Parameter	Description
	newly added since the last measurement period;
	<p>If a few lines have been added to a log file that was monitored previously, then the eG agent monitors the additions to that log file, and then proceeds to monitor newer log files (if any). If an older log file has been overwritten, then, the eG agent monitors this log file completely, and then proceeds to monitor the newer log files (if any).</p>
SearchPattern	<p>Enter the specific patterns of alerts to be monitored. The pattern should be in the following format: <i><PatternName>:<Pattern></i>, where <i><PatternName></i> is the pattern name that will be displayed in the monitor interface and <i><Pattern></i> is an expression of the form - <i>*expr*</i> or <i>expr</i> or <i>*expr</i> or <i>expr*</i>, etc. A leading '*' signifies any number of leading characters, while a trailing '**' signifies any number of trailing characters.</p> <p>For example, say you specify <i>ServerProcess:PSAPPSRV.7948*</i> in the SearchPattern text box. This indicates that "ServerProcess" is the pattern name to be displayed in the monitor interface. "PSAPPSRV.7948*" indicates that the test will monitor only those lines in the alert log which begin with the phrase PSAPPSRV.7948. Similarly, if your pattern specification reads: <i>IpAction:*null</i>, then it means that the pattern name is "IpAction" and that the test will monitor those lines in the alert log which end with the term "null".</p> <p>A single pattern may also be of the form <i>e1+e2</i>, where + signifies an OR condition. That is, the <i><PatternName></i> is matched if either <i>e1</i> is true or <i>e2</i> is true.</p> <p>Multiple search patterns can be specified as a comma-separated list. For example: <i>ServerProcess:PSAPPSRV.7948*,IpAction:*null</i></p> <p>If the AlertFile specification is of the format <i>Name@logfilepatterns</i>, then the descriptor for this test in the eG monitor interface will be of the format: <i>Name:PatternName</i>. On the other hand, if the AlertFile specification consists only of a comma-separated list of log files/patterns, then the descriptors will be of the format: <i>LogFile/Pattern:PatternName</i>.</p> <p>If you want all the messages in a log file to be monitored, then your specification would be: <i><PatternName>:*</i>.</p>
Lines	<p>Specify two numbers in the format <i>x:y</i>. This means that when a line in the alert file matches a particular pattern, then <i>x</i> lines before the matched line and <i>y</i> lines after the matched line will be reported in the detail diagnosis output (in addition to the matched line). The default value here is <i>0:0</i>. Multiple entries can be provided as a comma-separated list.</p> <p>If you give <i>1:1</i> as the value for Lines, then this value will be applied to all the patterns specified in the SearchPattern field. If you give <i>0:0,1:1</i> as the value for Lines and if the corresponding value in the SearchPattern field is like</p>

Parameter	Description
	<p><i>ServerProcess:PSAPPSRV.7948*,IpAction:*null</i>, then:</p> <p>0:0 will be applied to ServerProcess:PSAPPSRV.7948* pattern</p> <p>1:1 will be applied to IpAction:*nullpattern</p>
ExcludePattern	Provide a comma-separated list of patterns to be excluded from monitoring in the ExcludePattern text box. For example: <i>PSAPPSRV.19826*,PSAPPSRV.7949*</i> . By default, this parameter is set to ' <i>none</i> '.
UniqueMatch	<p>By default, the UniqueMatch parameter is set to False, indicating that, by default, the test checks every line in the log file for the existence of each of the configured SearchPatterns. By setting this parameter to True, you can instruct the test to ignore a line and move to the next as soon as a match for one of the configured patterns is found in that line. For example, assume that</p> <p><i>ServerProcess:*PSAPPSRV.7948*,IpAction:*null*</i> is the SearchPattern that has been configured. If UniqueMatch is set to False, then the test will read every line in the log file completely to check for the existence of messages embedding the strings '<i>PSAPPSRV.7948</i>' and '<i>null</i>'. If both the patterns are detected in the same line, then the number of matches will be incremented by 2. On the other hand, if UniqueMatch is set to True, then the test will read a line only until a match for one of the configured patterns is found and not both. This means that even if the strings '<i>PSAPPSRV.7948</i>' and '<i>null</i>' follow one another in the same line, the test will consider only the first match and not the next. The match count in this case will therefore be incremented by only 1.</p>
RotatingFile	<p>This flag governs the display of descriptors for this test in the eG monitoring console.</p> <p>If this flag is set to True, then the descriptors of this test will be displayed in the following format: <i>Directory_containing_monitored_file:<SearchPattern></i>. For instance, if the AlertFile parameter is set to <i>C:\ps\appserv\psdomain\LOGS\APPSRV_0415.log</i>, and RotatingFile is set to True, then, your descriptor will be of the following format: <i>C:\ps\appserv\psdomain\LOGS:<SearchPattern></i>. On the other hand, if the RotatingFile flag had been set to False, then the descriptors will be of the following format: <i><FileName>:<SearchPattern></i> - i.e., <i>APPSRV_0415.LOG:<SearchPattern></i> in the case of the example above.</p>
CaseSensitive	This flag is set to No by default. This indicates that the test functions in a 'case-insensitive' manner by default. This implies that, by default, the test ignores the case of your AlertFile and SearchPattern specifications. If this flag is set to Yes on the other hand, then the test will function in a 'case-sensitive' manner. In this case therefore, for the test to work, even the case of your AlertFile and SearchPattern specifications should match with the actuals.
RolloverFile	By default, this flag is set to False . Set this flag to True if you want the test to support

Parameter	Description
	<p>the ‘roll over’ capability (if any) of the specified AlertFile. A roll over typically occurs when the timestamp of a file changes or when the log file size crosses a pre-determined threshold. When a log file rolls over, the errors/warnings that pre-exist in that file will be automatically copied to a new file, and all errors/warnings that are captured subsequently will be logged in the original/old file. In such a scenario, since the RolloverFile flag is set to False by default, the test by default scans only the original/old file for new log entries and ignores the new file. On the other hand, if the flag is set to True, then the test will scan both old and the rolled-over file for new entries.</p> <p>If you want this test to support the ‘roll over’ capability described above, the following conditions need to be fulfilled:</p> <ul style="list-style-type: none"> • The AlertFile parameter has to be configured only with the name and/or path of one/more alert files. File patterns should not be specified in the AlertFile text box. • The roll over file name should be of the format: “<AlertFile>.1”, and this file must be in the same directory as the AlertFile.
OverwrittenFile	<p>By default, this flag is set to False. Set this flag to True if log files do not ‘roll over’ in your environment, but get overwritten instead. In such environments typically, new messages that are captured will be written into the log file that pre-exists and will replace the original contents of that log file; unlike when ‘roll over’ is enabled, no new log files are created for new entries in this case. If the OverwrittenFile flag is set to True, then the test will scan the new entries in the log file for matching patterns. However, if the flag is set to False, then the test will ignore the new entries.</p>
EncodeFormat	<p>By default, this is set to <i>none</i>, indicating that no encoding format applies by default. However, if the test has to use a specific encoding format for reading from the specified AlertFile, then you will have to provide a valid encoding format here. Where multiple log files are being monitored, you will have to provide a comma-separated list of encoding formats – one each for every log file monitored -- eg., UTF-8, UTF-16. Make sure that your encoding format specification follows the same sequence as your AlertFile specification. In other words, the first encoding format should apply to the first alert file, and so on.</p> <p>Note:</p> <p>If your AlertFile specification consists of file patterns that include wildcard characters (eg., C:\ps\appserv\psdomain\LOGS\TUXLOG.03*15, C:\ps\appserv\psdomain\LOGS\TUXLOG.04*15), then such configurations will only be supported in the ANSI format, and not the UTF format.</p>
UseUTF8	By default, this flag is set to No , implying that the test does not use the UTF-8 format

Parameter	Description
	<p>to read from the log files configured for monitoring. However, if the test has to use only the UTF-8 format for reading from all the log files configured for monitoring in your AlertFile specification, then set this flag to Yes.</p>
UseUTF16	<p>By default, this flag is set to No, implying that the test does not use the UTF-16 format to read from the log files configured for monitoring. However, if the test has to use only the UTF-16 format for reading from all the log files con figured for monitoring in your AlertFile specification, then set this flag to Yes.</p> <p>Note:</p> <p>If your AlertFile specification consists of file patterns that include wildcard characters (eg.,C:\ps\appserv\psdomain\LOGS\TUXLOG.03*15, C:\ps\appserv\psdomain\LOGS\TUXLOG.04*15), then such configurations will only be supported in the ANSI format, and not the UTF format.</p>
DD Frequency	<p>Refers to the frequency with which detailed diagnosis measures are to be generated for this test. The default is 1:1. This indicates that, by default, detailed measures will be generated every time this test runs, and also every time the test detects a problem. You can modify this frequency, if you so desire. Also, if you intend to disable the detailed diagnosis capability for this test, you can do so by specifying none against DD Frequency.</p>
Detailed Diagnosis	<p>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.</p> <p>The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled:</p> <ul style="list-style-type: none"> • 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
Number of messages	Indicates the number of messages matching this	Number	Use the detailed diagnosis of this measure to view these error

Measurement	Description	Measurement Unit	Interpretation
	SearchPattern found in this AlertFile.		messages completely.

The detailed diagnosis of the *Number of messages* measure displays the detailed messages that match the configured SearchPattern.

Figure 3.3: The detailed diagnosis of the Tuxedo PIA Application Logs Test

3.1.2 PeopleSoft Tuxedo Log Monitor Test

The **TUXLOG** files enable administrators to trace the Tuxedo component for troubleshooting information. Using the **PeopleSoft Tuxedo Log Monitor** Test, administrators can periodically scan these log files for specific patterns of errors/warnings, and can efficiently troubleshoot them.

Target of the test : A Tuxedo Application Server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for every AlertFile and SearchPattern pair configured.

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The host for which the test is being configured.
Port	Specify the port at which the specified host listens in the Port text box. By default, this is <i>NULL</i> .
AlertFile	This test monitors the TUXLOG files for errors/warnings. Typically, these log files will be available in the <PSHome>\appserv\<domain>\LOGS directory and will be named in the

Parameter	Description
	<p>following format: TUXLOG.mmddyy, where <mmddyy> refers to the month, date, and year of creation of the log file. In the logs directory therefore, you will find multiple TUXLOG files, one for every day of the month. To enable the test to monitor a specific TUXLOG file, you need to provide the full path to that log file against AlertFile. For instance, if <PSHome> (on Windows) is C:\ps, the name of the PeopleSoft domain is <i>psdomain</i>, and the log file you want to monitor was created on the 3rd of May, 2017, your AlertFile specification will be: C:\ps\appserv\psdomain\LOGS\TUXLOG.050317. Multiple log files can also be monitored as a comma-separated list. For instance, your AlertFile specification can be:</p> <p>C:\ps\appserv\psdomain\LOGS\TUXLOG.050317, C:\ps\appserv\psdomain\LOGS\TUXLOG.052017. Specific log file name patterns can also be specified. For example, to monitor all the log files created in the months of March and April (i.e., in the 3rd and 4th months), the parameter specification can be, :</p> <p>C:\ps\appserv\psdomain\LOGS\TUXLOG.03*15,C:\ps\appserv\psdomain\LOGS\TUXLOG.04*15. Here, '*' indicates leading/trailing characters (as the case may be).</p> <p>Your AlertFile specification can also be of the following format: <i>Name@logfilepath_or_pattern</i>. Here, <i>Name</i> represents the display name of the path being configured. For instance, to monitor all TUXLOG log files that were created in the months of March and April, your specification can be: <i>Marchlogs@ C:\ps\appserv\psdomain\LOGS\TUXLOG_03*15, Aprillogs@C:\ps\appserv\psdomain\LOGS\TUXLOG_04*15</i>. In this case, the display names 'Marchlogs' and 'Aprillogs' will alone be displayed as descriptors of this test.</p> <p>Every time this test is executed, the eG agent verifies the following:</p> <ul style="list-style-type: none"> • Whether any changes have occurred in the size and/or timestamp of the log files that were monitored during the last measurement period; • Whether any new log files (that match the AlertFile specification) have been newly added since the last measurement period; <p>If a few lines have been added to a log file that was monitored previously, then the eG agent monitors the additions to that log file, and then proceeds to monitor newer log files (if any). If an older log file has been overwritten, then, the eG agent monitors this log file completely, and then proceeds to monitor the newer log files (if any).</p> <p>SearchPattern Enter the specific patterns of alerts to be monitored. The pattern should be in the following format: <PatternName>:<Pattern>, where <PatternName> is the pattern name that will be displayed in the monitor interface and <Pattern> is an expression of the form - *expr* or expr or *expr or expr*, etc. A leading '*' signifies any number of leading characters, while a trailing '*' signifies any number of trailing characters.</p> <p>For example, say you specify <i>ORA:ORA-*</i> in the SearchPattern text box. This indicates that "ORA" is the pattern name to be displayed in the monitor interface. "ORA-**" indicates that the</p>

Parameter	Description
	<p>test will monitor only those lines in the alert log which start with the term "ORA-". Similarly, if your pattern specification reads: <i>offline</i>:*<i>offline</i>, then it means that the pattern name is offline and that the test will monitor those lines in the alert log which end with the term <i>offline</i>.</p> <p>A single pattern may also be of the form <i>e1+e2</i>, where + signifies an OR condition. That is, the <i><PatternName></i> is matched if either <i>e1</i> is true or <i>e2</i> is true.</p> <p>Multiple search patterns can be specified as a comma-separated list. For example:</p> <p><i>ORA:ORA-*,offline:*offline*,online:*online</i></p> <p>If the AlertFile specification is of the format <i>Name@/filepath</i>, then the descriptor for this test in the eG monitor interface will be of the format: <i>Name:PatternName</i>. On the other hand, if the AlertFile specification consists only of the log file path, then the descriptors will be of the format: <i>LogFilepath:PatternName</i>.</p> <p>If you want all the messages in a log file to be monitored, then your specification would be: <i><PatternName>:*</i>.</p>
Lines	<p>Specify two numbers in the format <i>x:y</i>. This means that when a line in the alert file matches a particular pattern, then <i>x</i> lines before the matched line and <i>y</i> lines after the matched line will be reported in the detail diagnosis output (in addition to the matched line). The default value here is <i>0:0</i>. Multiple entries can be provided as a comma-separated list.</p> <p>If you give <i>1:1</i> as the value for Lines, then this value will be applied to all the patterns specified in the SearchPattern field. If you give <i>0:0,1:1</i> as the value for Lines and if the corresponding value in the SearchPattern field is like <i>Error:ERROR:*</i>, <i>Fail:*failed</i>, then:</p> <p><i>0:0</i> will be applied to <i>Error:ERROR:*</i> pattern</p> <p><i>1:1</i> will be applied to <i>Fail:*failed</i> pattern</p>
ExcludePattern	<p>Provide a comma-separated list of patterns to be excluded from monitoring in the ExcludePattern text box. For example: <i>JOLT_CAT*,LIBTUX_CAT*</i>. By default, this parameter is set to '<i>none</i>'.</p>
UniqueMatch	<p>By default, this parameter is set to False, indicating that, by default, the test checks every line in the log file for the existence of each of the configured SearchPatterns. By setting this parameter to True, you can instruct the test to ignore a line and move to the next as soon as a match for one of the configured patterns is found in that line. For example, assume that <i>Error:*ERROR:*, Fail:*failed*</i> is the SearchPattern that has been configured. If UniqueMatch is set to False, then the test will read every line in the log file completely to check for the existence of messages embedding the strings 'ERROR:' and 'failed'. If both the patterns are detected in the same line, then the number of matches will be incremented by 2. On the other hand, if UniqueMatch is set to True, then the test will read a line only until a match for one of the configured patterns is found and not both. This means that even if the</p>

Parameter	Description
	strings 'ERROR:' and 'failed' follow one another in the same line, the test will consider only the first match and not the next. The match count in this case will therefore be incremented by only 1.
RotatingFile	<p>This flag governs the display of descriptors for this test in the eG monitoring console.</p> <p>If this flag is set to True, then the descriptors of this test will be displayed in the following format: <i>Directory_containing_monitored_file:<SearchPattern></i>. For instance, if the AlertFile parameter is set to C:\ps\appserv\psdomain\LOGS\TUXLOG.042015, and RotatingFile is set to True, then, your descriptor will be of the following format: C:\ps\appserv\psdomain\LOGS:<SearchPattern>. On the other hand, if the RotatingFile flag had been set to False, then the descriptors will be of the following format: <FileName>:<SearchPattern> - i.e., TUXLOG.042015:<SearchPattern> in the case of the example above.</p>
CaseSensitive	<p>This flag is set to No by default. This indicates that the test functions in a 'case-insensitive' manner by default. This implies that, by default, the test ignores the case of your AlertFile and SearchPattern specifications. If this flag is set to Yes on the other hand, then the test will function in a 'case-sensitive' manner. In this case therefore, for the test to work, even the case of your AlertFile and SearchPattern specifications should match with the actuals.</p>
RolloverFile	<p>By default, this flag is set to False. Set this flag to True if you want the test to support the 'roll over' capability (if any) of the specified AlertFile. A roll over typically occurs when the timestamp of a file changes or when the log file size crosses a pre-determined threshold. When a log file rolls over, the errors/warnings that pre-exist in that file will be automatically copied to a new file, and all errors/warnings that are captured subsequently will be logged in the original/old file. In such a scenario, since the RolloverFile flag is set to False by default, the test by default scans only the original/old file for new log entries and ignores the new file. On the other hand, if the flag is set to True, then the test will scan both old and the rolled-over file for new entries.</p> <p>If you want this test to support the 'roll over' capability described above, the following conditions need to be fulfilled:</p> <ul style="list-style-type: none"> • The AlertFile parameter has to be configured only with the name and/or path of one/more alert files. File patterns should not be specified in the AlertFile text box. • The roll over file name should be of the format: "<AlertFile>.1", and this file must be in the same directory as the AlertFile.
OverwrittenFile	<p>By default, this flag is set to False. Set this flag to True if log files do not 'roll over' in your environment, but get overwritten instead. In such environments typically, new messages that are captured will be written into the log file that pre-exists and will replace the original contents</p>

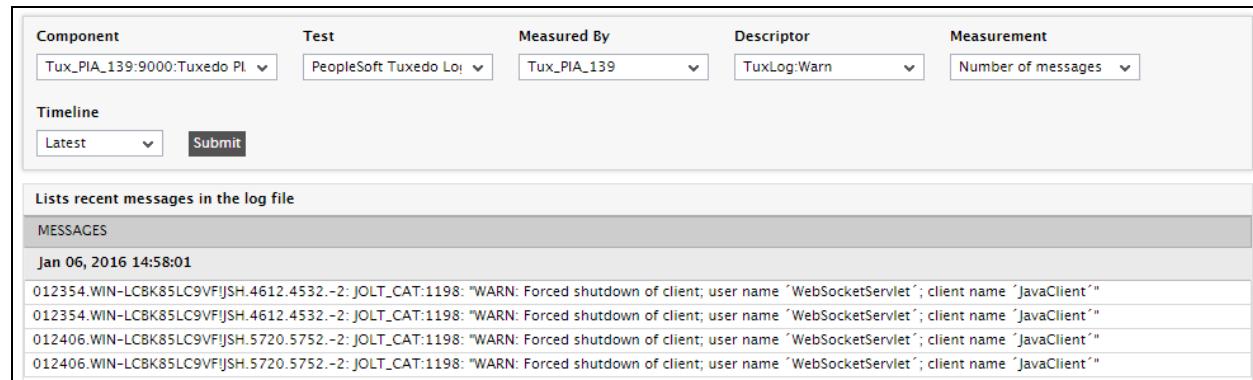
Parameter	Description
	of that log file; unlike when 'roll over' is enabled, no new log files are created for new entries in this case. If the OverwrittenFile flag is set to True , then the test will scan the new entries in the log file for matching patterns. However, if the flag is set to False , then the test will ignore the new entries.
EncodeFormat	By default, this is set to <i>none</i> , indicating that no encoding format applies by default. However, if the test has to use a specific encoding format for reading from the specified AlertFile, then you will have to provide a valid encoding format here. Where multiple log files are being monitored, you will have to provide a comma-separated list of encoding formats – one each for every log file monitored – eg., UTF-8, UTF-16. Make sure that your encoding format specification follows the same sequence as your AlertFile specification. In other words, the first encoding format should apply to the first alert file, and so on.
	Note: If your AlertFile specification consists of file patterns that include wildcard characters (eg., C:\ps\appserv\psdomain\LOGS\TUXLOG.03*15, C:\ps\appserv\psdomain\LOGS\TUXLOG.04*15), then such configurations will only be supported in the ANSI format, and not the UTF format.
UseUTF8	By default, this flag is set to No , implying that the test does not use the UTF-8 format to read from the log files configured for monitoring. However, if the test has to use only the UTF-8 format for reading from all the log files configured for monitoring in your AlertFile specification, then set this flag to Yes .
UseUTF16	By default, this flag is set to No , implying that the test does not use the UTF-16 format to read from the log files configured for monitoring. However, if the test has to use only the UTF-16 format for reading from all the log files configured for monitoring in your AlertFile specification, then set this flag to Yes .
	Note: If your AlertFile specification consists of file patterns that include wildcard characters (eg., C:\ps\appserv\psdomain\LOGS\TUXLOG.03*15, C:\ps\appserv\psdomain\LOGS\TUXLOG.04*15), then such configurations will only be supported in the ANSI format, and not the UTF format.
DD Frequency	Refers to the frequency with which detailed diagnosis measures are to be generated for this test. The default is <i>1:1</i> . This indicates that, by default, detailed measures will be generated every time this test runs, and also every time the test detects a problem. You can modify this frequency, if you so desire. Also, if you intend to disable the detailed diagnosis capability for this test, you can do so by specifying <i>none</i> against DD Frequency.
Detailed Diagnosis	To make diagnosis more efficient and accurate, the eG Enterprise suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run

Parameter	Description
<p>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.</p> <p>The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled:</p> <ul style="list-style-type: none"> • 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
Number of messages	Indicates the number of messages matching this SearchPattern found in this AlertFile.	Number	Use the detailed diagnosis of this measure to view these error messages completely.

The detailed diagnosis of the *Number of messages* measure displays the detailed messages that match the configured SearchPattern.



The screenshot shows the 'PeopleSoft Tuxedo Log Monitor Test' interface. At the top, there are dropdown menus for 'Component' (Tux_PIA_139:9000:Tuxedo Pl.), 'Test' (PeopleSoft Tuxedo Log Monitor), 'Measured By' (Tux_PIA_139), 'Descriptor' (TuxLog:Warn), and 'Measurement' (Number of messages). Below this is a 'Timeline' section with a dropdown set to 'Latest' and a 'Submit' button. The main area displays a list of log messages under the heading 'MESSAGES'. The messages are timestamped and show repeated entries of 'WARN' level log entries indicating forced shutdown of clients due to WebSocketServlet issues.

Timestamp	Log Message
Jan 06, 2016 14:58:01	012354.WIN-LCBK85LC9VFJSH.4612.4532.-2: JOLT_CAT:1198: "WARN: Forced shutdown of client; user name 'WebSocketServlet'; client name 'JavaClient'"
	012354.WIN-LCBK85LC9VFJSH.4612.4532.-2: JOLT_CAT:1198: "WARN: Forced shutdown of client; user name 'WebSocketServlet'; client name 'JavaClient'"
	012406.WIN-LCBK85LC9VFJSH.5720.5752.-2: JOLT_CAT:1198: "WARN: Forced shutdown of client; user name 'WebSocketServlet'; client name 'JavaClient'"
	012406.WIN-LCBK85LC9VFJSH.5720.5752.-2: JOLT_CAT:1198: "WARN: Forced shutdown of client; user name 'WebSocketServlet'; client name 'JavaClient'"

Figure 3.4: The detailed diagnosis of the PeopleSoft Tuxedo Log Monitor Test

3.2 The Tuxedo PIA Server Layer

With the help of the tests mapped to this layer, administrators can:

- Know how many request queues the monitored domain consists of, and which queue has the maximum number of enqueued requests;
- Identify the server process that is unable to process requests quickly, and is hence enqueueing many requests in its queue;
- Determine the current workload of the domain and the clients who are contributing to this load;
- Ascertain the current status of the server processes in the domain.

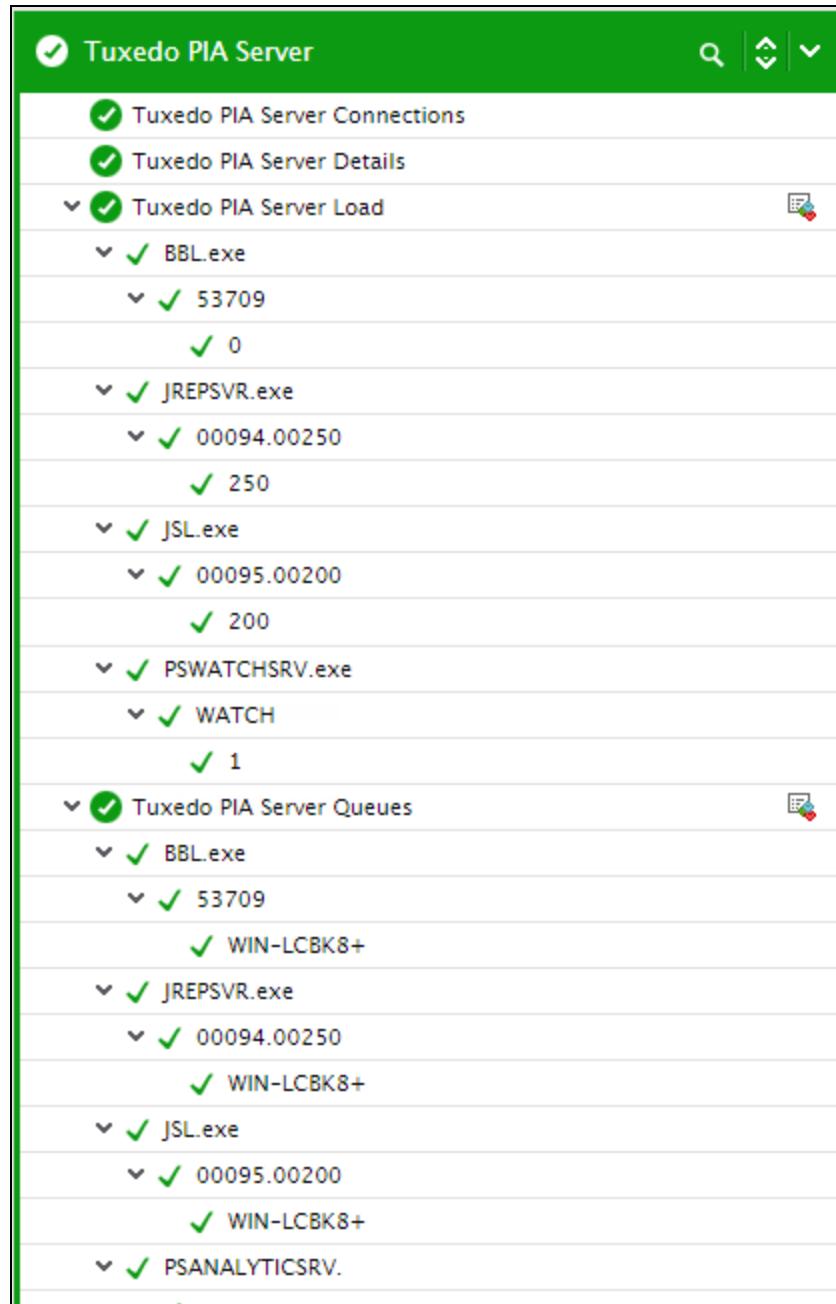


Figure 3.5: The tests mapped to the Tuxedo PIA Server layer

3.2.1 Tuxedo PIA Server Load Test

Typically, when configuring an application server, administrators can set the “weight” of the load, or amount of requests, that need to be directed to a particular application server. Weight values are integers 1–10, with 1 being low and 10 being a heavy load. Servers that can handle extra work can

take heavy loads, while servers that are either less powerful or are being used in other capacities can take lower loads.

When monitoring the queues used by each node of an application server domain, it is not only important to know the number of requests enqueued in each queue at any given point in time, but also how this request count translates into “weighted load” on the node.

By measuring the weighted load on each queue used by every server process on a domain node, administrators can analyze how load-intensive every queue is, and isolate that queue that imposes the maximum load on the node. This is where the **Tuxedo PIA Server Load** test helps. This test auto-discovers the queues used by every server process in a domain node, and reports the requests handled and the weighted load per queue. In the process, the test pinpoints that queue which contributes the most to the load on the node, and thus leads administrators to the type of transactions that are being the most load-intensive.

Target of the test : A Tuxedo Application Server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for each queue used by every server process in each node of an application server domain

First-level descriptor: Server process

Second-level descriptor: Queue ID

Third-level descriptor: Machine ID that represents the node of the application server domain. In a partitioned domain, where a single domain exists across more than one node, different partitions or nodes within a domain must be given different names. The PeopleSoft configuration is delivered as a single node domain by default. However, it is perfectly possible to configure it as a partitioned domain.

Configurable parameters for the test

Parameter	Description
Test period	How often should the test be executed
Host	The host for which the test is to be configured.
Port	The port number at which the specified Host listens to. By default, this will be 12345.
PSAdmin Home	To collect metrics from a Tuxedo application server, this test runs PeopleSoft Server Administration (PSADMIN) commands on the target server. The first step towards this

Parameter	Description
	<p>is to launch the psadmin.exe. To enable the test to run the psadmin.exe, you need to configure the test with the full path to the directory in which the psadmin.exe resides. Therefore, in the PSAdmin Home text box, specify the location of the psadmin.exe. Typically, this will be the install directory of the Tuxedo application server. For instance, on Windows, your specification can be C:\ps\appsrv and on Unix, your specification can be: <i>opt/ps/appsrv</i>.</p>
Domain	<p>An application server Domain is the collection of server processes, supporting processes, and resource managers that enable connections to the database. A single application server machine can support multiple application server domains running on it. A server process is executable code that receives incoming transaction requests. The server process carries out a request by making calls to a service. Using a managed Tuxedo application server, you can monitor only those server processes and transactions that pertain to a single domain. This is why, the eG agent needs to be explicitly configured with the application server Domain it needs to monitor.</p>

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Requests handled	Indicates the number of requests currently handled by this queue.	Number	<p>This is a good indicator of the current queue length. A consistent rise in the value of this measure is a cause for concern, as it indicates that many requests to the corresponding server process are pending processing. This hints at a probable processing bottleneck in the server.</p>
Server load	Indicates the weighted load imposed by this queue on the server.	Number	<p>This is a weighted measure of the server requests. Some requests could have a different weight than others. By default, the workload is always 50 times the number of requests.</p> <p>Compare the value of this measure across queues to know which queue is imposing the maximum load on the domain.</p>
Request rate	Indicates the rate at which requests were handled by	Requests/Sec	

Measurement	Description	Measurement Unit	Interpretation
	this queue.		
Load rate	Indicates the load rate of this server queue.	Load/Sec	

3.2.2 Tuxedo PIA Client Connections Test

Monitoring the clients that are communicating with the Tuxedo application server domain and their experience with the domain, will accurately point administrators to clients who are imposing the maximum load on the domain and the clients whose experience with the domain is below-par. This is exactly what the **Tuxedo PIA Client Connections** test does. This test automatically discovers the clients who are currently connected to the server domain, reports the number of active sessions for each client, and also reveals the number of transactions of every client that have been committed and aborted. In the process, the test pinpoints those clients who have launched too many sessions on the domain; in the event that the server domain chokes, such clients could be considered guilty of overloading the domain. Additionally, the test reveals which client's transactions were aborted often, rendering such clients unable to complete their transactions; this in turn may disrupt the business flow and adversely impact experience with the server domain.

Target of the test : A Tuxedo Application Server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for every client currently connected to the server domain 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 number at which the specified Host listens to. By default, this will be 12345.
PSAdmin Home	To collect metrics from a Tuxedo application server, this test runs PeopleSoft Server Administration (PSADMIN) commands on the target server. The first step towards this is to launch the psadmin.exe . To enable the test to run the psadmin.exe , you need to configure the test with the full path to the directory in which the psadmin.exe resides. Therefore, in the PSAdmin Home text box, specify the location of the psadmin.exe .

Parameter	Description
	<p>Typically, this will be the install directory of the Tuxedo application server. For instance, on Windows, your specification can be C:\ps\appsrv and on Unix, your specification can be: <i>opt/ps/appsrv</i>.</p>
Domain	<p>An application server Domain is the collection of server processes, supporting processes, and resource managers that enable connections to the database. A single application server machine can support multiple application server domains running on it. A server process is executable code that receives incoming transaction requests. The server process carries out a request by making calls to a service. Using a managed Tuxedo application server, you can monitor only those server processes and transactions that pertain to a single domain. This is why, the eG agent needs to be explicitly configured with the application server Domain it needs to monitor.</p>
Detailed Diagnosis	<p>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.</p> <p>The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled:</p> <ul style="list-style-type: none"> • 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
Number of clients	Indicates the number of sessions currently active for this client.	Number	<p>This is a good indicator of the workload currently imposed by a client on the application server domain. In the event of an overload, you can compare the value of this measure across clients to know which client is contributing to the overload by launching many sessions on the domain.</p> <p>You can also use the detailed</p>

Measurement	Description	Measurement Unit	Interpretation
			diagnosis of this measure to know which users are communicating with the application server domain via the client, the duration of each user's session with the domain, and the current status of the session. With the help of this information, administrators can in no time isolate those sessions that are idle, but are unnecessarily consuming resources and adding to the server load. By closing such sessions, administrators can conserve server resources and reduce load.
Average begins	Indicates the average number of transactions started by this client.	Number	
Average commits	Indicates the average number of transactions committed by this client.	Number	
Average aborts	Indicates the average number of transactions of this client that were aborted.	Number	Ideally, the value of this measure should be low. A high value could indicate the incidence of many transactional errors that are causing the transactions to abort. You may want to compare the value of this measure across clients to know which client's transactions aborted the most, thus leading administrators to those clients who may not have had a good experience with the server.

The detailed diagnosis of the *Number of clients* measure reveals the users who are communicating with the app server domain, the duration of each user's session with the domain, and the current status of the session. Idle sessions can thus be identified and their impact on resource consumption can be assessed.

Component	Test	Measured By	Descriptor	Measurement		
Tux_PIA_139:9000:Tuxedo PI	Tuxedo PIA Client Cor	Tux_PIA_139	192.168.10.97	Number of clients		
Timeline						
<input type="button" value="Latest"/> <input type="button" value="Submit"/>						
Details of Top 10 clients based on duration						
LMID	USER NAME	DURATION(MINS)	STATUS	BEGIN	COMMIT	ABORTED
Jan 06, 2016 14:53:32						
WIN-LCBK85LC9VF	JPOOL_25	280	IDLE/W	0	0	0
WIN-LCBK85LC9VF	JPOOL_26	149	IDLE/W	0	0	0

Figure 3.6: Detailed diagnosis of the Tuxedo PIA Client Connections test

3.2.3 Tuxedo PIA Server Details Test

Using this test, administrators can figure out how many server processes in the domain are currently servicing requests, how many are active and ready for requests but have not been assigned requests yet, and the number of server processes that are not even ready for requests. With the help of the detailed diagnosis of this test, administrators can swiftly identify which server process is in which state presently, the number of requests being handled by each server process currently, and the total number of requests per server process for which processing is complete. Using this information, administrators can determine the following:

- Which server processes are idle?
- Which server processes are active, but are not processing any request?
- Which server process is handling too many requests currently?

Target of the test : A Tuxedo Application Server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for the application server domain 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 number at which the specified Host listens to. By default, this will be 12345.
PSAdmin Home	To collect metrics from a Tuxedo application server, this test runs PeopleSoft Server Administration (PSADMIN) commands on the target server. The first step towards this

Parameter	Description
	<p>is to launch the psadmin.exe. To enable the test to run the psadmin.exe, you need to configure the test with the full path to the directory in which the psadmin.exe resides. Therefore, in the PSAdmin Home text box, specify the location of the psadmin.exe. Typically, this will be the install directory of the Tuxedo application server. For instance, on Windows, your specification can be C:\ps\appsrv and on Unix, your specification can be: <i>opt/ps/appsrv</i>.</p>
Domain	<p>An application server Domain is the collection of server processes, supporting processes, and resource managers that enable connections to the database. A single application server machine can support multiple application server domains running on it. A server process is executable code that receives incoming transaction requests. The server process carries out a request by making calls to a service. Using a managed Tuxedo application server, you can monitor only those server processes and transactions that pertain to a single domain. This is why, the eG agent needs to be explicitly configured with the application server Domain it needs to monitor.</p>
Detailed Diagnosis	<p>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.</p> <p>The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled:</p> <ul style="list-style-type: none"> • 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
Active servers	Indicates the number of server processes that are active, but are not processing any request currently.	Number	Use the detailed diagnosis of this measure to know which are the active server processes.
Running servers	Indicates the number of server processes that are	Number	Use the detailed diagnosis of this measure to know which server

Measurement	Description	Measurement Unit	Interpretation
	actively servicing requests presently.		processes are running currently.
Idle servers	Indicates the number of server processes that are not even ready for servicing any requests currently.	Number	Use the detailed diagnosis of this measure to know which are the idle server processes.
Other servers	Indicates the number of server processes that are in any other state.	Number	Use the detailed diagnosis of this measure to know which server processes are in a state other than Active, Running, or Idle.

To identify the idle server processes, use the detailed diagnosis of the *Idle servers* measure.

Component	Test	Measured By	Measurement	Timeline		
Tux_PIA_139:9000:Tuxedo Pl.	Tuxedo PIA Server Det	Tux_PIA_139	Idle servers	Latest	<input type="button" value="Submit"/>	
Details of Idle servers						
PROGRAM NAME	QUEUE NAME	GROUP NAME	ID	STATUS	REQUEST HANDLED	LOAD DONE
Jan 06, 2016 14:54:49						
PSDBGSRV.exe	DBCQ	DBCSRVR	1	IDLE	0	0
JSL.exe	95.002	JSLGRP	200	IDLE	0	0
PSRENSRV.exe	RENQ1	RENGRP	101	IDLE	0	0
PSAMSRSV.exe	SAMQ	APPSRV	100	IDLE	0	0
PSAPPSRV.exe	APPQ	APPSRV	2	IDLE	121	6050
PSAPPSRV.exe	APPQ	APPSRV	1	IDLE	0	0
PSWATCHSRV.exe	WATCH	WATCH	1	IDLE	0	0
PSANALYTICSRV.	80	ANALYTIC+	3	IDLE	0	0
BBL.exe	53709	WIN-LCB+	0	IDLE	134	6700
PSANALYTICSRV.	80	ANALYTIC+	2	IDLE	0	0

Figure 3.7: The detailed diagnosis of the Idle servers measure of the Tuxedo PIA Server Details test

3.2.4 Tuxedo PIA Database Connections Test

If the Tuxedo application 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 the **Tuxedo PIA Database Connections** test, administrators can periodically monitor the usage of each connection pool in every logical group configured in an application server domain, and promptly detect when a pool runs short of connections. This way, the test provides useful pointers to resizing connection pools.

Target of the test : A Tuxedo Application Server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for every connection pool of each logical group in the application server domain being monitored.

First-level descriptor: Group name. Application servers in a domain are logically divided into groups.

Second-level descriptor: Connection pool

Configurable parameters for the test

Parameter	Description
Test period	How often should the test be executed
Host	The host for which the test is to be configured.
Port	The port number at which the specified Host listens to. By default, this will be 12345.
PSAdmin Home	To collect metrics from a Tuxedo application server, this test runs PeopleSoft Server Administration (PSADMIN) commands on the target server. The first step towards this is to launch the psadmin.exe . To enable the test to run the psadmin.exe , you need to configure the test with the full path to the directory in which the psadmin.exe resides. Therefore, in the PSAdmin Home text box, specify the location of the psadmin.exe . Typically, this will be the install directory of the Tuxedo application server. For instance, on Windows, your specification can be C:\ps\appsrv and on Unix, your specification can be: <i>opt/ps/appsrv</i> .
Domain	An application server Domain is the collection of server processes, supporting processes, and resource managers that enable connections to the database. A single application server machine can support multiple application server domains running on it. A server process is executable code that receives incoming transaction requests. The server process carries out a request by making calls to a service. Using a managed Tuxedo application server, you can monitor only those server processes and transactions that pertain to a single domain. This is why, the eG agent needs to be explicitly configured with the application server Domain it needs to monitor.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Max connection pool size	Indicates the maximum number of connections in	Number	

Measurement	Description	Measurement Unit	Interpretation
	this pool.		
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 the pool.	Number	
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.

3.3 The Tuxedo PIA Service Layer

Using the tests mapped to this layer, administrators can determine the current configuration of the domain, the load on the conversational server in the domain, the status of transactions to the domain, the length of the queues in the domain, and the status and load of the services in the domain.

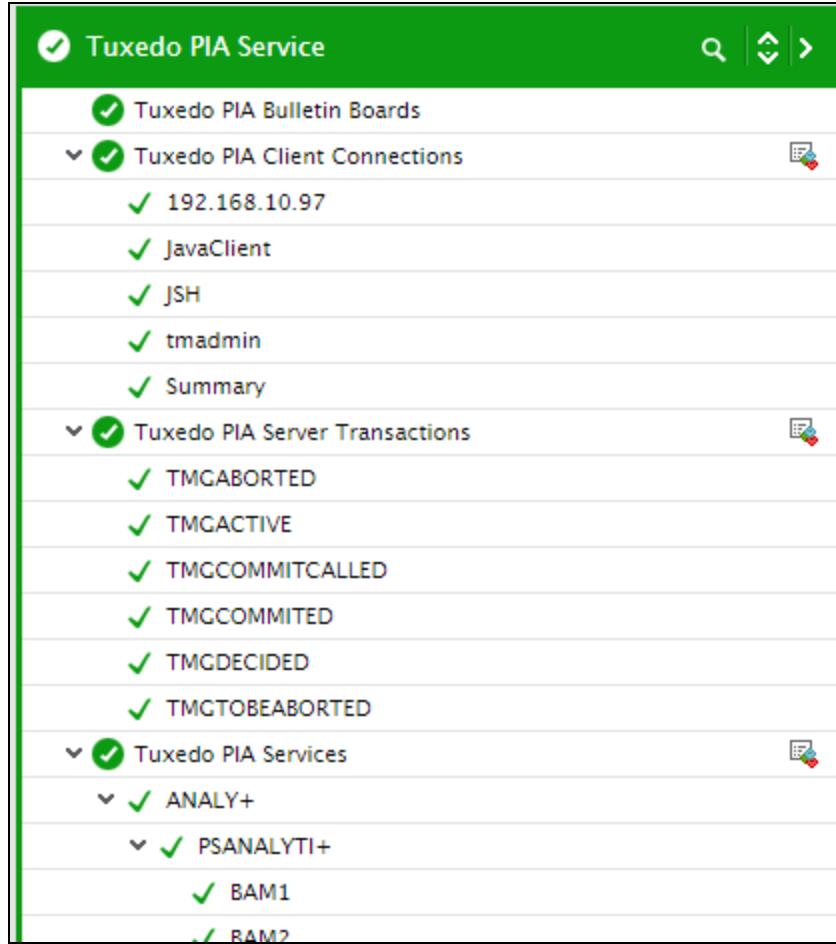


Figure 3.8: The tests mapped to the Tuxedo PIA Service layer

3.3.1 Tuxedo PIA Bulletin Boards Test

When any Tuxedo application server domain is booted, the first process to be started is the Bulletin Board Liaison process, called BBL. This process is the heart of the application server. It reads the configuration of the domain from a binary configuration file, which in a PeopleSoft domain is called PSTUXCFG. The BBL then establishes a shared memory segment, referred to as the Bulletin Board (BB) or Management Information Base (MIB), some message queues (determined by the specifications set out in the configuration file), and two semaphores. The Bulletin Board holds all the information about the rest of the application server domain including servers, services and their loads and priorities, clients and transactions. It is used as a form of database by the application server processes to determine how they should behave.

By monitoring how requests to a domain are utilizing the server processes, services, and interfaces registered with the Bulletin Board, administrators can ascertain the load on the domain, and whether the domain has been sized right to handle the load. This is exactly what the **Tuxedo PIA Bulletin**

Boards test helps administrators achieve. This test reports the current configuration of the domain by reading the bulletin board. In addition, the test also keeps an eye on the usage of the server processes, services, and interfaces that the domain has been configured with, and proactively alerts administrators to probable processing bottlenecks in the domain, along with accurate pointers to the cause of the bottleneck – is it because the domain does not have enough server processes? services? Or interfaces? Based on the test's findings, administrators can then proceed to reconfigure the domain (if required) to ensure rapid processing of requests.

Target of the test : A Tuxedo Application Server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for the application server domain 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 number at which the specified Host listens to. By default, this will be 12345.
PSAdmin Home	To collect metrics from a Tuxedo application server, this test runs PeopleSoft Server Administration (PSADMIN) commands on the target server. The first step towards this is to launch the psadmin.exe . To enable the test to run the psadmin.exe , you need to configure the test with the full path to the directory in which the psadmin.exe resides. Therefore, in the PSAdmin Home text box, specify the location of the psadmin.exe . Typically, this will be the install directory of the Tuxedo application server. For instance, on Windows, your specification can be C:\ps\appsrv and on Unix, your specification can be: <i>opt/ps/appsrv</i> .
Domain	An application server Domain is the collection of server processes, supporting processes, and resource managers that enable connections to the database. A single application server machine can support multiple application server domains running on it. A server process is executable code that receives incoming transaction requests. The server process carries out a request by making calls to a service. Using a managed Tuxedo application server, you can monitor only those server processes and transactions that pertain to a single domain. This is why, the eG agent needs to be explicitly configured with the application server Domain it needs to monitor.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Total servers	Indicates the total number of server processes registered with the bulletin board.	Number	A server process is an executable code that receives and processes incoming transaction requests. The number of server processes of each type is configurable and typically varies within a domain, depending on the requirements of your application or the main purpose of the domain. To gauge the adequacy of the server process configuration, you need to use the <i>Used Servers</i> and <i>Free servers</i> measures.
Used servers	Indicates the number of server processes that are actively processing requests.	Number	If the Used servers count is very high and the percentage of Free servers is low, it implies that the load on the domain is high, but the domain has not been configured with sufficient server processes to handle the load. If the percentage of free server processes is allowed to dip further, request queues on the domain will only grow longer, delaying request processing or even bringing it to a standstill. To avoid such adversities, you have to probe deeper to identify the server process types that are in demand and configure more server processes of that type in the domain.
Free servers	Indicates the percentage of server processes that are currently idle.	Percent	The server process carries out a request by making calls to a service, such as <i>MgrGetObject</i> . The server process waits for the service to complete, then returns information to the device that initiated the request, such as a browser.
Total services	Indicates the total number of services registered with the bulletin board.	Number	A high value for the <i>Used services</i>

Measurement	Description	Measurement Unit	Interpretation
			measure indicates that many services are in use, which in turn implies that the load on the domain is high. If the percentage of Free services is also low, it indicates that the domain has not been configured with adequate services to complete the requests. In such situations, you will have to consider increasing the number of services configured for the domain.
Used services	Indicates the number of services that are actively processing requests to the domain.	Number	
Free services	Indicates the percentage of services idle in the domain.	Percent	
Total interfaces	Indicates the number of network interfaces registered with the bulletin board.	Number	
Used interfaces	Indicates the number of network interfaces currently in use in the domain.	Number	
Free interfaces	Indicates the percentage of network interfaces idle in the domain.	Percent	A high value is desired for this measure. If the value of this measure is low and the value of the <i>Used interfaces</i> measure is close to that of the <i>Total interfaces</i> measure, it indicates that the domain has very few interfaces left for handling any additional traffic. Under such circumstances, you may want to assign more network interfaces to the domain.
Number of queues	Indicates the current number of queues in the domain.	Number	While a server process waits for a service to complete, other transaction requests wait in a queue until the current service completes. A service may take a fraction of a second to complete or several seconds, depending on the type and complexity of the service. When the service

Measurement	Description	Measurement Unit	Interpretation
			<p>completes, the server process is then available to process the next request in the corresponding queue.</p> <p>The existence of multiple queues in a domain could be sign of a processing bottleneck in the domain, caused by poor domain configuration. If the value of this measure is high therefore, you may want to check the value of the Free servers, Free services, and Free interfaces, to figure out what resource the domain requires to ensure that there are no hiccups in request processing.</p>
Number of groups	Indicates the number of groups currently available in the domain.	Number	Application servers in a domain are logically divided into groups. These groups permit the same service to be configured differently on different servers in different groups.

3.3.2 Tuxedo PIA Server Connections Test

The PSSAMSRV application server process running within an application server domain is what Tuxedo refers to as a conversational server. This server process is capable of conversing with other servers by submitting requests to them, or receiving requests from them. It provides only one service, SqlAccess. This server is used to administer the process scheduler tables, and to allocate version numbers on PeopleSoft objects during development and upgrade. When a job is submitted, or when job and process scheduler statuses are updated in the process monitor, the SQL to perform those updates are submitted directly by PSSAMSRV.

By tracking connections to the PSSAMSRV application server process in a domain over time, an administrator can assess the typical load on the server process. Based on this assessment, the administrators can, if required, fine-tune the settings of the PSSAMSRV so as to enhance its request-handling capability. This is where the Tuxedo PIA Server Connections test helps. This test periodically reports the current connection load on the PSSAMSRV application server process. By using this test to monitor how the load on PSSAMSRV varies with time, administrators can gauge the typical load on the PSSAMSRV application server process and can accordingly figure out

whether/not additional PSSAMSRV application server process instances will be required for handling the load. Additionally, administrators can use the detailed diagnosis of this test to know which two server processes are conversing using the conversational server process in the domain.

Target of the test : A Tuxedo Application Server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for the application server domain 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 number at which the specified Host listens to. By default, this will be 12345.
PSAdmin Home	To collect metrics from a Tuxedo application server, this test runs PeopleSoft Server Administration (PSADMIN) commands on the target server. The first step towards this is to launch the psadmin.exe . To enable the test to run the psadmin.exe , you need to configure the test with the full path to the directory in which the psadmin.exe resides. Therefore, in the PSAdmin Home text box, specify the location of the psadmin.exe . Typically, this will be the install directory of the Tuxedo application server. For instance, on Windows, your specification can be C:\ps\appsrv and on Unix, your specification can be: <i>opt/ps/appsrv</i> .
Domain	An application server Domain is the collection of server processes, supporting processes, and resource managers that enable connections to the database. A single application server machine can support multiple application server domains running on it. A server process is executable code that receives incoming transaction requests. The server process carries out a request by making calls to a service. Using a managed Tuxedo application server, you can monitor only those server processes and transactions that pertain to a single domain. This is why, the eG agent needs to be explicitly configured with the application server Domain it needs to monitor.
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:

Parameter	Description
	<ul style="list-style-type: none"> • 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	Descriptions	Measurement Unit	Interpretation
No of connections	Indicates the number of conversational connections currently open on the Tuxedo server.	Number	<p>This is a good indicator of the current workload of the PSSAMSRV application server process. To know the details of the current connections, use the detailed diagnosis of this test. This reveals the Originator group ID, the Original logical machine ID, Receiver group ID and logical machine ID, Data transmitted by originator and receiver, and service name. With the help of the detailed metrics, administrators can identify which originator-receiver pair is transmitting the maximum data.</p> <p>If the value of this measure increases consistently, and the PSSAMSRV application server process appears to service requests slowly, it could be because enough server instances are not available to handle the heavy load.</p> <p>It is mandatory that at least one instance of the PSSAMSRV application server process be started when a domain is booted. Depending upon the variations in the value of this measure over time, you may want to increase the number of instances to be started at boot time, so that the load is balanced and serviced quickly.</p>

3.3.3 Tuxedo PIA Server Queues Test

When a Tuxedo application server process carries out a request, it makes a call to a service, such as MgrGetObject. The server process waits for the service to complete, then returns information to the device that initiated the request, such as a browser. While a server process waits for a service to complete, other transaction requests wait in a queue until the current service completes. A service may take a fraction of a second to complete or several seconds, depending on the type and complexity of the service. When the service completes, the server process is then available to process the next request in the corresponding queue.

If the length of a queue keeps increasing, it is a sign that the corresponding server process is unable to process requests quickly. Therefore, to be able to swiftly spot a probable processing bottleneck in an application server domain, administrators must monitor the queues in the domain continuously, isolate those queues that are growing in length, and identify which server process is using that queue. This can be achieved using the **Tuxedo PIA Server Queues** test. This test auto-discovers the queues used by each server process in every node of a domain, and for every queue, reports the number of requests in the queue and the number of instances of that server process sharing the queue. This way, the test rapidly leads administrators to a potential slowdown in request processing by the domain node, and helps them accurately isolate which server process is contributing to the slowdown.

Target of the test : A Tuxedo Application Server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for every queue used by each server process of a domain node

First-level descriptor: Server process

Second-level descriptor: Queue ID

Third-level descriptor: Machine ID that represents the node of the application server domain. In a partitioned domain, where a single domain exists across more than one node, different partitions or nodes within a domain must be given different names. The PeopleSoft configuration is delivered as a single node domain by default. However, it is perfectly possible to configure it as a partitioned domain.

Configurable parameters for the test

Parameter	Description
Test period	How often should the test be executed
Host	The host for which the test is to be configured.
Port	The port number at which the specified Host listens to. By default, this will be 12345.
PSAdmin Home	To collect metrics from a Tuxedo application server, this test runs PeopleSoft Server Administration (PSADMIN) commands on the target server. The first step towards this is to launch the psadmin.exe . To enable the test to run the psadmin.exe , you need to configure the test with the full path to the directory in which the psadmin.exe resides. Therefore, in the PSAdmin Home text box, specify the location of the psadmin.exe . Typically, this will be the install directory of the Tuxedo application server. For instance, on Windows, your specification can be C:\ps\appsrv and on Unix, your specification can be: <i>opt/ps/appsrv</i> .
Domain	An application server Domain is the collection of server processes, supporting processes, and resource managers that enable connections to the database. A single application server machine can support multiple application server domains running on it. A server process is executable code that receives incoming transaction requests. The server process carries out a request by making calls to a service. Using a managed Tuxedo application server, you can monitor only those server processes and transactions that pertain to a single domain. This is why, the eG agent needs to be explicitly configured with the application server Domain it needs to monitor.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Servers connected	Indicates the number of instances of this server process that are currently connected to this queue.	Number	This is a good indicator of the number of instances of a server processes that are currently running in a domain.
Average queue length	Indicates the average length of this queue.	Number	A consistent rise in the value of this measure is a cause for concern, as it indicates that this server process is not able to process the requests in this queue quickly. Under such circumstances, you may need to add more instances of that particular server

Measurement	Description	Measurement Unit	Interpretation
			process to improve its processing capability.
Service requests queued	Indicates the number of service requests in this queue.	Number	
All requests in queue	Indicates the total number of requests in this queue.	Number	

3.3.4 Tuxedo PIA Services Test

Server processes invoke services to perform application logic and issue SQL to the RDBMS. When a PeopleSoft application sends a request to the application server in a group, it sends a service name and a set of parameters, such as MgrGetObject and its parameters. Tuxedo then queues the transaction request to a specific server process that is designed to handle certain services. When a server process boots, it advertises to the system the predefined services it handles.

If the server process is down or is not ready to handle any requests, then all services it handles will also be unavailable; this can severely hamper request processing by the server. Moreover, if even a single service processes requests slowly, user experience with the application server will be adversely impacted. Using the **Tuxedo PIA Services** test, administrators can quickly detect such anomalies and initiate measures to correct them, so that requests are serviced promptly. Additionally, the test also periodically reports the count of requests handled by each service, thus highlighting the most popular/busiest services delivered by the application server domain.

Target of the test : A Tuxedo Application Server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for every service invoked by each server process in every application server group in a domain.

First-level descriptor: Application server group; Application servers in a domain are logically divided into groups. These groups permit the same service to be configured differently on different servers in different groups.

Second-level descriptor: Server process name

Third-level descriptor: Service name

Configurable parameters for the test

Parameter	Description
Test period	How often should the test be executed
Host	The host for which the test is to be configured.
Port	The port number at which the specified Host listens to. By default, this will be 12345.
PSAdmin Home	To collect metrics from a Tuxedo application server, this test runs PeopleSoft Server Administration (PSADMIN) commands on the target server. The first step towards this is to launch the psadmin.exe . To enable the test to run the psadmin.exe , you need to configure the test with the full path to the directory in which the psadmin.exe resides. Therefore, in the PSAdmin Home text box, specify the location of the psadmin.exe . Typically, this will be the install directory of the Tuxedo application server. For instance, on Windows, your specification can be C:\ps\appsrv and on Unix, your specification can be: <i>opt/ps/appsrv</i> .
Domain	An application server Domain is the collection of server processes, supporting processes, and resource managers that enable connections to the database. A single application server machine can support multiple application server domains running on it. A server process is executable code that receives incoming transaction requests. The server process carries out a request by making calls to a service. Using a managed Tuxedo application server, you can monitor only those server processes and transactions that pertain to a single domain. This is why, the eG agent needs to be explicitly configured with the application server Domain it needs to monitor.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation						
Service status	Indicates the current status of this service.		<p>The values that this measure can report and their corresponding numeric values are discussed in the table below:</p> <table border="1"> <thead> <tr> <th>Measure Value</th> <th>Numeric Value</th> </tr> </thead> <tbody> <tr> <td>Available</td> <td>100</td> </tr> <tr> <td>Not available</td> <td>0</td> </tr> </tbody> </table> <p>Note:</p>	Measure Value	Numeric Value	Available	100	Not available	0
Measure Value	Numeric Value								
Available	100								
Not available	0								

Measurement	Description	Measurement Unit	Interpretation
			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 since 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 services are most popular or are the busiest.
Request rate	Indicates the rate at which requests were handled by this service.	Requests/Sec	A consistent dip in the value of this measure is a cause for concern as it indicates that the service is unable to process requests quickly. This could hint at a processing slowdown.

3.3.5 Tuxedo PIA Server Transactions Test

A Tuxedo application server consists of numerous PeopleSoft server processes, grouped in domains. Each server process within a domain provides unique processing abilities, enabling the application server to respond effectively to a multitude of transaction requests generated throughout the PeopleSoft architecture.

When a PeopleSoft application sends a request to the application server, it sends a service name and a set of parameters, such as MgrGetObject and its parameters. The application server then queues the transaction request to a specific server process that is designed to handle certain services. The server process then establishes a persistent connection to a PeopleSoft database, and this connection acts as a generic SQL pipeline that the server process uses to send and receive SQL to process the transaction request.

If any of the transaction requests handled by a domain fail, administrators should be able to proactively detect the failure and prevent it from occurring, so that workflow is not disrupted and user productivity is not hit. The **Tuxedo PIA Server Transactions** test helps administrators with this and

more! This test periodically checks the status of transactions on a configured application server domain, and reports how many transactions are in which state presently. This can reveal the number of active transactions on the domain, based on which the current load on the domain can be determined. In addition, the test also warns administrators of transactional errors that may cause transactions to abort, thus enabling administrators to take pre-emptive steps in this regard.

Target of the test : A Tuxedo Application Server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for every transaction status code returned by the application server.

Configurable parameters for the test

Parameter	Description
Test period	How often should the test be executed
Host	The host for which the test is to be configured.
Port	The port number at which the specified Host listens to. By default, this will be 12345.
PSAdmin Home	To collect metrics from a Tuxedo application server, this test runs PeopleSoft Server Administration (PSADMIN) commands on the target server. The first step towards this is to launch the psadmin.exe . To enable the test to run the psadmin.exe , you need to configure the test with the full path to the directory in which the psadmin.exe resides. Therefore, in the PSAdmin Home text box, specify the location of the psadmin.exe . Typically, this will be the install directory of the Tuxedo application server. For instance, on Windows, your specification can be C:\ps\appsrv and on Unix, your specification can be: <i>opt/ps/appsrv</i> .
Domain	An application server Domain is the collection of server processes, supporting processes, and resource managers that enable connections to the database. A single application server machine can support multiple application server domains running on it. A server process is executable code that receives incoming transaction requests. The server process carries out a request by making calls to a service. Using a managed Tuxedo application server, you can monitor only those server processes and transactions that pertain to a single domain. This is why, the eG agent needs to be explicitly configured with the application server Domain it needs to monitor.
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,

Parameter	Description
	<p>choose the On option. To disable the capability, click on the Off option.</p> <p>The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled:</p> <ul style="list-style-type: none"> • 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 that are in this status currently.	Number	<p>If the value of this measure is high for the tmactive descriptor, it implies that many transactions are currently active on the domain. You may want to observe variations to this load over time, so that, if demand appears to be increasing, you can consider configuring multiple application server domains per database, for redundancy, fail-over, and performance reasons.</p> <p>On the other hand, if the value of this measure is very high for the tmgaborted or the tmgtobeaborted descriptors, it means that many transactions have been aborted or are about to be aborted. This is a cause for concern, and requires further investigation.</p> <p>Use the detailed diagnosis of this measure to know the details of transactions in various states. This way, you can accurately pinpoint the transactions that are about to be aborted.</p>

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.

Copyright © 2018 eG Innovations Inc. All rights reserved.

This document may not be reproduced by any means nor modified, decompiled, disassembled, published or distributed, in whole or in part, or translated to any electronic medium or other means without the prior written consent of eG Innovations. eG Innovations makes no warranty of any kind with regard to the software and documentation, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The information contained in this document is subject to change without notice.

All right, title, and interest in and to the software and documentation are and shall remain the exclusive property of eG Innovations. All trademarks, marked and not marked, are the property of their respective owners. Specifications subject to change without notice.