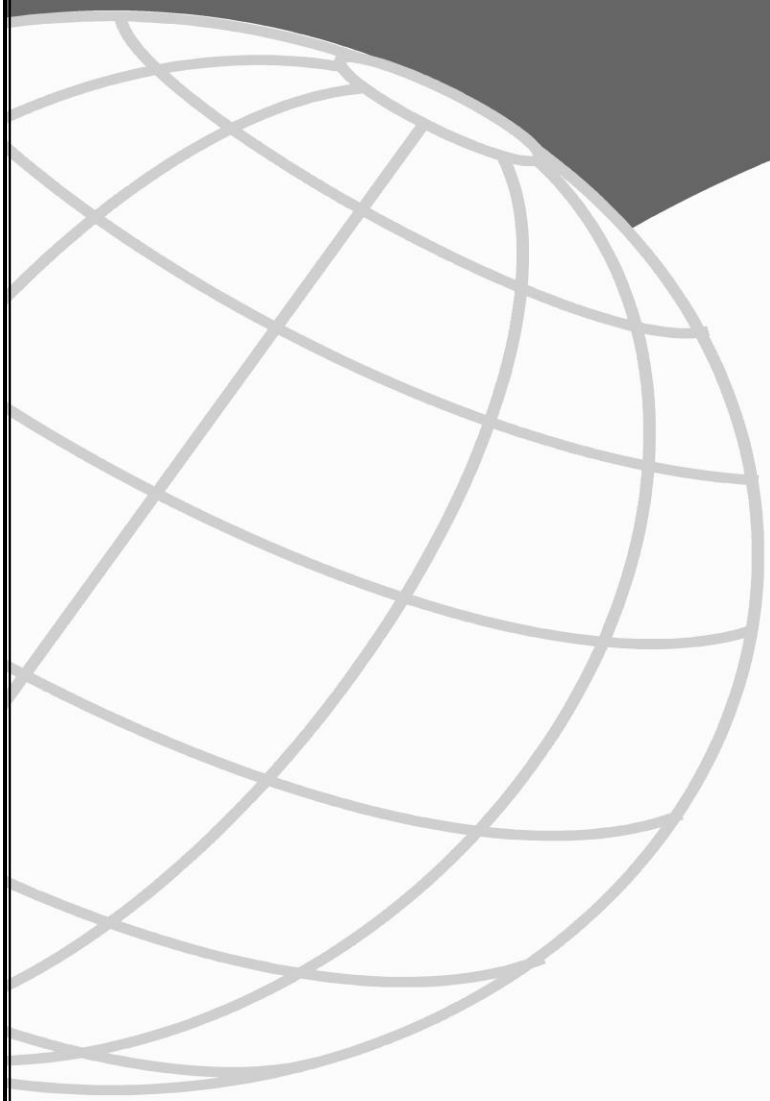

Additional Tests

eG Enterprise v6

Restricted Rights Legend

The information contained in this document is confidential and subject to change without notice. No part of this document may be reproduced or disclosed to others without the prior permission of eG Innovations Inc. eG Innovations Inc. 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.

Trademarks

Microsoft Windows, Windows NT, Windows 2000, Windows 2003 and Windows 2008 are either registered trademarks or trademarks of Microsoft Corporation in United States and/or other countries.

The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

Copyright

Table of Contents

Chapter 1

Additional Tests

The eG Enterprise suite provides for a few in-built tests that can be associated with any existing server type or new server type that is added using the Integration Console utility.

Note:

The tests discussed in this document will not be available for any of the existing (i.e. built-in) server types. If need be, you can associate one/more of these tests to an existing server-type/layer using the licensed **eG Integration Console** component.

1.1 Process Pools Test

The ProcessPools test reports a variety of CPU and memory statistics pertaining to every process in a process tree, starting from the root-process to its leaves (i.e. it reports measures related to both parent and child processes). The measures made by this test are as follows:

Purpose	Reports a variety of CPU and memory statistics pertaining to every process in a process tree, starting from the root-process to its leaves
Target of the test	
Agent deploying the test	An Internal agent

<table>
<tr><td>Configurable parameters for the test</td><td colspan="3">1. TEST PERIOD - How often should the test be executed
2. PROCESS - Enter a comma separated list of names:pattern pairs which identify the process(es) associated with the server being considered. processName is a string that will be used for display purposes only. processPattern is an expression of the form - *expr* or expr or *expr or expr* or *expr1*expr2*... or expr1*expr2, etc. A leading '*' signifies any number of leading characters, while a trailing '*' signifies any number of trailing characters. For example, for an iPlanet application server (Nas_server), there are three processes named kcs, kjs, and kxs associated with the application server. For this server type, in the PROCESS text box, enter "kcsProcess:*kcs*, kjsProcess:*kjs*, kxsProcess:*kxs*, where * denotes zero or more characters. Other special characters such as slashes (\) can also be used while defining the process pattern. For example, if a server's root directory is /home/egurkha/apache and the server executable named httpd exists in the bin directory, then, the process pattern is "*/home/egurkha/apache/bin/httpd*".
To determine the process pattern to use for your application, on Windows environments, look for the process name(s) in the Task Manager -> Processes selection. To determine the process pattern to use on Unix environments, use the ps command (e.g., the command "ps -e -o pid,args" can be used to determine the processes running on the target system; from this, choose the processes of interest to you).
3. PIDFILE - Enter a comma separated list of process names:paths to pid files that contain the process ids of the processes that need to be monitored. processName is a string that will be used for display purposes only. For example, this text box could contain, WebServer:/tmp/pid_file1, Apache:/tmp/pid_file2, where pid_file1 and pid_file2 are the files containing the process ids. Note that each pid file can contain only one pid.</td></tr>
<tr><td>Outputs of the test</td><td colspan="3">One set of results for the server being monitored</td></tr>
<tr><td>Measurements made by the test</td><td>Measurement</td><td>Measurement Unit</td><td>Interpretation</td></tr>
<tr><td></td><td>Processes running:
Number of instances of a process(es) currently executing on a host</td><td>Number</td><td>This value indicates if too many or too few processes corresponding to an application are executing on the host.</td></tr>
<tr><td></td><td>CPU usage:
Percentage of CPU used by executing process(es) corresponding to the pattern specified</td><td>Percent</td><td>A very high value could indicate that processes corresponding to the specified pattern are consuming excessive CPU resources.</td></tr>
</table>

	Memory usage: For one or more processes corresponding to a specified set of patterns, this value represents the ratio of the resident set size of the processes to the physical memory of the host system, expressed as a percentage.	Percent	A sudden increase in memory utilization for a process(es) may be indicative of memory leaks in the application.

Note:

If a log file to be monitored is not found or is empty, then the errcount will be 0.

1.2 Tcp Connection Test

This test various statistics pertaining to TCP connections to and from a host, from an external perspective.

Purpose	Tracks various statistics pertaining to TCP connections to and from a host, from an external perspective.
Target of the test	
Agent deploying the test	An external agent
Configurable parameters for the test	1. **TEST PERIOD** - How often should the test be executed 2. **HOST** - Host name of the server for which the test is to be configured 3. **PORTNO** - Enter the port to which the specified **HOST** listens 4. **TARGETPORTS** – Specify either a comma-separated list of port numbers that are to be tested (eg., 80,7077,1521), or a comma-separated list of *port name:port number* pairs that are to be tested (eg., smtp:25,mssql:1433). In the latter case, the port name will be displayed in the monitor interface. Alternatively, this parameter can take a comma-separated list of *port name:IP address:port number* pairs that are to be tested, so as to enable the test to try and connect to Tcp ports on multiple IP addresses. For example, *mysql:192.168.0.102:1433,egwebsite:209.15.165.127:80*. 5. **ISPASSIVE** – If the value chosen is **YES**, then the server under consideration is a passive server in a cluster. No alerts will be generated if the server is not running. Measures will be reported as "Not applicable' by the agent if the server is not up.

Outputs of the test	One set of results for every configured port name		
Measurements made by the test	**Measurement**	**Measurement Unit**	**Interpretation**
	Availability: Whether the TCP connection is available	Percent	An availability problem can be caused by different factors – e.g., the server process may not be up, a network problem may exist, or there could be a configuration problem with the DNS server.
	Response time: Time taken (in seconds) by the server to respond to a request.	Secs	An increase in response time can be caused by several factors such as a server bottleneck, a configuration problem with the DNS server, a network problem, etc.

1.3 Exception Log Test

The XceptionLog test reports general statistics pertaining to the log files in a host.

Purpose	Reports general statistics pertaining to the log files in a host
Target of the test	
Agent deploying the test	An Internal agent

<table>
<tr><td>Configurable parameters for the test</td><td colspan="3">1. TEST PERIOD - How often should the test be executed
2. HOST - Host name/IP address of the server for which the test is to be configured
3. PORTNO - The port on which the specified server listens for HTTP requests
4. LOGFILE - The name of the log file to be monitored
5. LOGDIR - The full path to the specified log file
6. EMPTYFILE - Enter either true or false. The entry true instructs the eG Enterprise suite to monitor even empty log files. The entry false instructs the eG Enterprise suite to ignore empty log files during monitoring. By default, this text box will hold the value false.
7. HIGHPATTERN - In order to track critical exceptions logged in the log file, you need to specify the pattern of such exceptions, here. For eg., if critical exception logs contain the string "Error", then your pattern specification could be *Error*. A leading '*' signifies any number of leading characters, while a trailing '*' signifies any number of trailing characters.
8. LOWPATTERN - To monitor minor exceptions logged in the log file, the pattern of the minor exceptions has to be specified in this text box. For eg., if minor exception logs contain the string "Low", then the pattern specification could be *Low*. A leading '*' signifies any number of leading characters, while a trailing '*' signifies any number of trailing characters.
9. MEDIUMPATTERN - For monitoring the medium exceptions in the log file, the pattern of these exceptions needs to be defined in this text box. For eg., if medium exception logs contain the string "Warning", then the pattern specification could be *Warning*. A leading '*' signifies any number of leading characters, while a trailing '*' signifies any number of trailing characters.</td></tr>
<tr><td>Outputs of the test</td><td colspan="3">One set of results for the server being monitored</td></tr>
<tr><td rowspan="4">Measurements made by the test</td><td>Measurement</td><td>Measurement Unit</td><td>Interpretation</td></tr>
<tr><td>Total exceptions:
Indicates the total number of exceptions logged in the log file</td><td>Number</td><td>A high value of this measure indicates the need to analyze the exceptions, ascertain their severity, and take corrective action if required.</td></tr>
<tr><td>High xceptions:
Indicates the number of critical exceptions that have been logged in the log file</td><td>Number</td><td>System performance will suffer much on the occurrence of critical exceptions. Such exceptions will have to be fixed with immediate effect.</td></tr>
<tr><td>Medium exceptions:
Indicates the number of not-very-critical exceptions logged in the log file</td><td>Number</td><td>Medium exceptions might not have an immediate impact on the system performance, but, in the long run, they could grow to be fatal. Such exceptions need not be looked into immediately, but will have to be fixed soon enough.</td></tr>
</table>

	Low exceptions: Indicates the number of very minor exceptions in the log file	Number	Low exceptions are very negligible in nature and can be ignored.

Note:

If a log file to be monitored is not found or is empty, then the errcount will be 0.

1.4 Message Log Test

The MsgLog test reports general statistics pertaining to the log files in a host.

Purpose	Reports general statistics pertaining to the log files in a host
Target of the test	
Agent deploying the test	An Internal agent

<table>
<tr><td>Configurable parameters for the test</td><td colspan="3">1. TEST PERIOD - How often should the test be executed
2. HOST - Host name/IP address of the server for which the test is to be configured
3. PORTNO - The port on which the specified server listens for HTTP requests
4. LOGFILE - The name of the log file to be monitored
5. LOGDIR - The full path to the specified log file
6. EMPTYFILE - Enter either true or false. The entry true instructs the eG Enterprise suite to monitor even empty log files. The entry false instructs the eG Enterprise suite to ignore empty log files during monitoring. By default, this text box will hold the value false.
7. HIGHPATTERN - In order to track critical exceptions logged in the log file, you need to specify the pattern of such exceptions, here. For eg., if critical exception logs contain the string "Error", then your pattern specification could be *Error*. A leading '*' signifies any number of leading characters, while a trailing '*' signifies any number of trailing characters.
8. LOWPATTERN - To monitor minor exceptions logged in the log file, the pattern of the minor exceptions has to be specified in this text box. For eg., if minor exception logs contain the string "Low", then the pattern specification could be *Low*. A leading '*' signifies any number of leading characters, while a trailing '*' signifies any number of trailing characters.
9. MEDIUMPATTERN - For monitoring the medium exceptions in the log file, the pattern of these exceptions needs to be defined in this text box. For eg., if medium exception logs contain the string "Warning", then the pattern specification could be *Warning*. A leading '*' signifies any number of leading characters, while a trailing '*' signifies any number of trailing characters.</td></tr>
<tr><td>Outputs of the test</td><td colspan="3">One set of results for the server being monitored</td></tr>
<tr><td rowspan="3">Measurements made by the test</td><td>Measurement</td><td>Measurement Unit</td><td>Interpretation</td></tr>
<tr><td>Number of exceptions:
Indicates the total number of exceptions logged in the log file</td><td>Number</td><td>A high value of this measure indicates the need to analyze the exceptions, ascertain their severity, and take corrective action if required.</td></tr>
<tr><td>High exception count:
Indicates the number of critical exceptions that have been logged in the log file</td><td>Number</td><td>System performance will suffer much on the occurrence of critical exceptions. Such exceptions will have to be fixed with immediate effect.</td></tr>
</table>

	Medium exception count: Indicates the number of not-very-critical exceptions logged in the log file	Number	Medium exceptions might not have an immediate impact on the system performance, but, in the long run, they could grow to be fatal. Such exceptions need not be looked into immediately, but will have to be fixed soon enough.
	Low exception count: Indicates the number of very minor exceptions in the log file	Number	Low exceptions are very negligible in nature and can be ignored.

Note:

If a log file to be monitored is not found or is empty, then the errcount will be 0.

1.5 Error Log Test

The ErrorLog test reports general statistics pertaining to the log files in a host.

Purpose	Reports general statistics pertaining to the log files in a host
Target of the test	
Agent deploying the test	An Internal agent
Configurable parameters for the test	1. **TEST PERIOD** - How often should the test be executed 2. **HOST** - Host name/IP address of the server for which the test is to be configured 3. **PORTNO** - The port on which the specified server listens for HTTP requests 4. **LOGFILE** - The name of the log file to be monitored 5. **LOGDIR** - The full path to the specified log file 6. **EMPTYFILE** - Enter either **true** or **false**. The entry **true** instructs the eG Enterprise suite to monitor even empty log files. The entry **false** instructs the eG Enterprise suite to ignore empty log files during monitoring. By default, this text box will hold the value **false**. 7. **ERRPATTERN** - In order to track the errors logged in a log file, you need to specify the pattern for the error logs in this text box. For eg., if the error logs contain the string "Error", then your pattern specification could be *Error*. A leading '*' signifies any number of leading characters, while a trailing '*' signifies any number of trailing characters.

Outputs of the test	One set of results for the server being monitored		
Measurements made by the test	**Measurement**	**Measurement Unit**	**Interpretation**
	Exceptions: Indicates the total number of errors logged in the log file	Number	A high value of this measure indicates an urgent need to identify the root-cause of the errors and take corrective action.

Note:

If a log file to be monitored is not found or is empty, then the errcount will be 0.

1.6 Process Details Test

This test is used to monitor the memory leaks (if any) in any Windows application or process. This test is particularly useful in development and staging environments, where memory leaks with applications can be detected early and recoding done to overcome the leaks.

Purpose	Monitors the memory leaks (if any) in any Windows application or process.		
Target of the test			
Agent deploying the test	An internal agent		
Configurable parameters for the test	1. **TEST PERIOD** - How often should the test be executed 2. **HOST** - The host for which the test is to be configured. 3. **PORT** – The port at which the server listens 4. **PROCESSNAME** - The name of the Windows application / process to be monitored. Multiple applications can be specified as a comma-separated list.		
Outputs of the test	One set of results for every process being monitored.		
Measurements made by the test	**Measurement**	**Measurement Unit**	**Interpretation**
	Current handles: Indicates the total number of file handles that are currently owned by each thread in the process.	Number	If there is a consistent increase in the value of this measure over time, then it is a clear indicator of a memory leak in the process.

	Private memory: Indicates the resources (handles, physical RAM, the paging file, system resources, etc.) that the process has allocated that cannot be shared with other processes.	KB	If there is a consistent increase in the value of this measure over time, then it is a clear indicator of a memory leak in the process.
	Pool paged memory usage: Indicates the memory in the paged pool. A paged pool is an area of system memory for objects that can be written to the disk, but which must remain in the physical memory.	KB	If there is a consistent increase in the value of this measure over time, then it is a clear indicator of a memory leak in the process.
	Pool non-paged memory usage: Indicates the memory in the non-paged pool. A non-paged pool is an area of system memory for objects that cannot be written to the disk, but which must remain in the physical memory as long as they are allocated.	KB	If there is a consistent increase in the value of this measure over time, then it is a clear indicator of a memory leak in the process.

1.7 Alert Log Test

This test monitors multiple alert log files for different patterns.

Purpose	Monitors multiple alert log files for different patterns
Target of the test	
Agent deploying the test	An internal agent

Configurable parameters for the test	1. **TEST PERIOD** - How often should the test be executed 2. **HOST** - The host for which the test is to be configured. 3. **PORT** – The port at which the server listens 4. **ALERTFILE -** Specify the path to the log file to be monitored. For eg., */user/john/new_john.log*. Multiple log file paths can be provided as a comma-separated list - eg., */user/john/critical_egurkha.log,/tmp/log/major.log*. Also, instead of a specific log file path, the path to the directory containing log files can be provided - eg., */user/logs*. This ensures that eG Enterprise monitors the most recent log files in the specified directory. Specific log file name patterns can also be specified. For example, to monitor the latest log files with names containing the strings 'dblogs' and 'applogs', the parameter specification can be, */tmp/db/*dblogs*,/tmp/app/*applogs**. Here, '*' indicates leading/trailing characters (as the case may be). In this case, the eG agent first enumerates all the log files in the specified path that match the given pattern, and then picks only the latest log file from the result set for monitoring. Your **ALERTFILE** specification can also be of the following format: *Name@logfilepath_or_pattern*. Here, *Name* represents the display name of the path being configured. Accordingly, the parameter specification for the 'dblogs' and 'applogs' example discussed above can be: *dblogs@/tmp/db/*dblogs*,applogs@/tmp/app/*applogs**. In this case, the display names 'dblogs' and 'applogs' will alone be displayed as descriptors of this test. **Note:** If your **ALERTFILE** specification consists of file patterns that include wildcard characters (eg., */tmp/db/*dblogs*,/tmp/app/*applogs**), then such configurations will only be supported in the ANSI format, and not the UTF format. Every time this test is executed, the eG agent verifies the following: ➢ Whether any changes have occurred in the size and/or timestamp of the log files that were monitoring during the last measurement period; ➢ Whether any new log files (that match the **ALERTFILE** specification) have been newly added since the last measurement period; 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).

5. **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.

   For example, say you specify ORA:ORA-* in the SEARCHPATTERN text box. This indicates that "ORA" is the pattern name to be displayed in the monitor interface. "ORA-*" indicates that the test will monitor only those lines in the alert log which start with the term "ORA-". Similarly, if your pattern specification reads: offline:*offline, 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 offline.

   A single pattern may also be of the form e1+e2, where + signifies an OR condition. That is, the *<PatternName>* is matched if either e1 is true or e2 is true.

   Multiple search patterns can be specified as a comma-separated list. For example: ORA:ORA-*,offline:*offline*,online:*online

   If the **ALERTFILE** specification is of the format *Name@logfilepath*, then the descriptor for this test in the eG monitor interface will be of the format: *Name:PatternName*. On the other hand, if the **ALERTFILE** specification consists only of a comma-separated list of log file paths, then the descriptors will be of the format: *LogFilePath:PatternName*.

   If you want all the messages in a log file to be monitored, then your specification would be: *<PatternName>:**.

6. **LINES** - Specify two numbers in the format x:y. This means that when a line in the alert file matches a particular pattern, then x lines before the matched line and y lines after the matched line will be reported in the detail diagnosis output (in addition to the matched line). The default value here is 0:0. Multiple entries can be provided as a comma-separated list.

   If you give 1:1 as the value for **LINES**, then this value will be applied to all the patterns specified in the **SEARCHPATTERN** field. If you give 0:0,1:1,2:1 as the value for **LINES** and if the corresponding value in the **SEARCHPATTERN** filed is like ORA:ORA-*,offline:*offline*,online:*online then:

   0:0 will be applied to ORA:ORA-* pattern

   1:1 will be applied to offline:*offline* pattern

   2:1 will be applied to online:*online pattern

	7. **EXCLUDEPATTERN** - Provide a comma-separated list of patterns to be excluded from monitoring in the **EXCLUDEPATTERN** text box. For example **critical*, *exception**. By default, this parameter is set to 'none'. 8. **UNIQUEMATCH** - 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 *Pattern1:*fatal*,Pattern2:*error** 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 'fatal' and 'error'. 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 'fatal' and 'error' 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.

9. **ROTATINGFILE** - This flag governs the display of descriptors for this test in the eG monitoring console.

   If this flag is set to **true** and the **ALERTFILE** text box contains the full path to a specific (log/text) file, then, the descriptors of this test will be displayed in the following format: *Directory_containing_monitored_file:<SearchPattern>*. For instance, if the **ALERTFILE** parameter is set to *c:\eGurkha\logs\syslog.txt*, and **ROTATINGFILE** is set to **true**, then, your descriptor will be of the following format: *c:\eGurkha\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., *syslog.txt:<SearchPattern>* in the case of the example above.

   If this flag is set to **true** and the **ALERTFILE** parameter is set to the directory containing log files, then, the descriptors of this test will be displayed in the format: *Configured_directory_path:<SearchPattern>.* For instance, if the **ALERTFILE** parameter is set to *c:\eGurkha\logs*, and **ROTATINGFILE** is set to **true**, then, your descriptor will be: *c:\eGurkha\logs:<SearchPattern>*. On the other hand, if the **ROTATINGFILE** parameter had been set to **false**, then the descriptors will be of the following format: *Configured_directory:<SearchPattern>* - i.e., *logs:<SearchPattern>* in the case of the example above.

   If this flag is set to true and the **ALERTFILE** parameter is set to a specific file pattern, then, the descriptors of this test will be of the following format: *<FilePattern>:<SearchPattern>*. For instance, if the **ALERTFILE** parameter is set to *c:\eGurkha\logs*sys**, and **ROTATINGFILE** is set to **true**, then, your descriptor will be: **sys*:<SearchPattern>*. In this case, the descriptor format will not change even if the **ROTATINGFILE** flag status is changed .**DD FREQUENCY** - Refers to the frequency with which detailed diagnosis measures are to be generated for this test. The default is *1:1*. This indicates that, by default, detailed measures will be generated every time this test runs, and also every time the test detects a problem. You can modify this frequency, if you so desire. Also, if you intend to disable the detailed diagnosis capability for this test, you can do so by specifying *none* against **DD FREQUENCY**.

10. **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.

11. **ROLLOVERFILE** - By default, this flag is set to **false**. Set this flag to **true** if you want the test to support the 'roll over' capability 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. For instance, say, errors and warnings were originally logged to a file named *error_log*. When a roll over occurs, the content of the file *error_log* will be copied to a file named *error_log.1*, and all new errors/warnings will be logged in *error_log*. In such a scenario, since the **ROLLOVERFILE** flag is set to **false** by default, the test by default scans only *error_log.1* for new log entries and ignores *error_log*. On the other hand, if the flag is set to **true**, then the test will scan both *error_log* and *error_log.1* for new entries.

    If you want this test to support the 'roll over' capability described above, the following conditions need to be fulfilled:

    - The **ALERTFILE** parameter has to be configured only with the name and/or path of one/more alert files. File patterns or directory specifications 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**.

12. **OVERWRITTENFILE** - 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 error/warning 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.

13. **ENCODEFORMAT** – By default, this is set to *none*, 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 - eg., *UTF-8, UTF-16*, etc. 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. 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. For instance, say that your alertfile specification is as follows: *D:\logs\report.log,E:\logs\error.log, C:\logs\warn_log*. Assume that while *UTF-8* needs to be used for reading from *report.log* , *UTF-16* is to be used for reading from *warn_log* . No encoding format need be applied to *error.*log. In this case, your **ENCODEFORMAT** specification will be: *UTF-8,none,UTF-16*.

    Note:

    If your **ALERTFILE** specification consists of file patterns that include wildcard characters (eg., */tmp/db/*dblogs*,/tmp/app/*applogs**), then such configurations will only be supported in the ANSI format, and not the UTF format.

	14. **DD FREQUENCY** - Refers to the frequency with which detailed diagnosis measures are to be generated for this test. The default is *1:1*. This indicates that, by default, detailed measures will be generated every time this test runs, and also every time the test detects a problem. You can modify this frequency, if you so desire. Also, if you intend to disable the detailed diagnosis capability for this test, you can do so by specifying *none* against **DD FREQUENCY**. 15. **DETAILED DIAGNOSIS** - To make diagnosis more efficient and accurate, the eG Enterprise suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the **On** option. To disable the capability, click on the **Off** option. The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled: ➢ The eG manager license should allow the detailed diagnosis capability ➢ Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.
Outputs of the test	One set of results for every **ALERTFILE** and **SEARCHPATTERN** combination

Measurements made by the test	Measurement	Measurement Unit	Interpretation
	Recent errors: Indicates the number of errors that were added to the alert log when the test was last executed.	Number	The value of this measure is a clear indicator of the number of "new" alerts that have come into the alert log of the monitored database. The detailed diagnosis of this measure, if enabled, provides the detailed descriptions of the errors of the configured patterns.

1.8 Directory Test

This test monitors one or more directories on a server.

Purpose	Monitors one or more directories on a server
Target of the test	
Agent deploying the test	An internal agent

Configurable parameters for the test	1. **TEST PERIOD** - How often should the test be executed 2. **HOST** - The host for which the test is to be configured. 3. **PORT** – The port at which the server listens 4. **TARGETDIRS** - Specify a comma-separated list of directory names to be monitored 5. **RECURSIVE** - This flag indicates if the test must check the target directories recursively or not. If this flag is set to **TRUE**, then all the sub-directories of each target directory are also checked.		
Outputs of the test	One set of results for every directory being monitored		
Measurements made by the test	**Measurement**	**Measurement Unit**	**Interpretation**
	Total files: Indicates the total number of files in a target directory.	Number	
	Total sub directories: Indicates the total number of sub-directories in a target directory.	Number	
	Modified files: Indicates the number of files in the target directory that were modified in the last measurement period.	Number	
	Directory size: Indicates the total size of all the files in the target directory.	MB	If the value of this measure is found to be alarmingly high, then ensure that unnecessary files occupying large amounts of directory space are immediately identified and removed. This is essential in order to ensure optimum use of the available disk space.

1.9 Old Files Test

This test tracks the age of the files within a specified directory on the system.

Purpose	Tracks the age of the files within a specified directory on the system
Target of the test	
Agent deploying the	An internal agent

test			
Configurable parameters for the test	1. **TEST PERIOD** - How often should the test be executed 2. **HOST** - The host for which the test is to be configured. 3. **PORT** – The port at which the server listens 4. **TARGETDIRS** - Specify the full path to the directory where the files to be monitored are created 5. **RECURSIVE** - If this flag is set to **TRUE**, then all the sub-directories of each target directory are also checked. 6. **MAXAGE** - This test will report the number of files that are older than the duration (in minutes) specified in the **MAXAGE** text box.		
Outputs of the test	One set of results for every directory being monitored.		
Measurements made by the test	**Measurement**	**Measurement Unit**	**Interpretation**
	Total files: The total number of files in the directory being monitored.	Number	
	Total old files: The total number of old files - i.e. the files for which last modified time was smaller than the current time.	Number	

1.10 File Size Test

The FileSize test monitors the file size of each of the files specified as parameters to the test.

Purpose	Tracks the age of the files within a specified directory on the system
Target of the test	
Agent deploying the test	An internal agent
Configurable parameters for the test	1. **TEST PERIOD** - How often should the test be executed 2. **HOST** - The host for which the test is to be configured. 3. **PORT** – The port at which the server listens 4. **FILES** - Specify a comma separated list of file reference and file path combinations - e.g., *agentlog:c:\eg\agent\logs\agentout.log,managerlog:c:\eg\manager\logs\error_log*.

Outputs of the test	One set of results for every file configured		
Measurements made by the test	**Measurement**	**Measurement Unit**	**Interpretation**
	Current size: The current size of the file in Kilobytes	KB	Alerts can be generated when a file exceeds a pre-defined maximum size.

1.11 Network Traps Test

The NetworkTraps test reports the count of SNMP trap messages sent on account of errors in the transactions between the network devices.

Purpose	Reports the count of SNMP trap messages sent on account of errors in the transactions between the network devices
Target of the test	An SNMP trap
Agent deploying the test	An internal agent

<table>
<tr><td>Configurable parameters for the test</td><td>
1. TEST PERIOD - How often should the test be executed
2. HOST - The host for which the test is to be configured
3. SOURCEADDRESS - Specify a comma-separated list of IP addresses or address patterns of the hosts sending the traps. For example, 10.0.0.1,192.168.10.*. A leading '*' signifies any number of leading characters, while a trailing '*' signifies any number of trailing characters.
4. OIDVALUE - Provide a comma-separated list of OID and value pairs returned by the traps. The values are to be expressed in the form, DisplayName:OID-OIDValue. For example, assume that the following OIDs are to be considered by this test: .1.3.6.1.4.1.9156.1.1.2 and .1.3.6.1.4.1.9156.1.1.3. The values of these OIDs are as given hereunder:
<table>
<tr><th>OID</th><th>Value</th></tr>
<tr><td>.1.3.6.1.4.1.9156.1.1.2</td><td>Host_system</td></tr>
<tr><td>.1.3.6.1.4.1.9156.1.1.3</td><td>NETWORK</td></tr>
</table>
In this case the OIDVALUE parameter can be configured as Trap1:.1.3.6.1.4.1.9156.1.1.2-Host_system,Trap2:.1.3.6.1.4.1.9156.1.1.3-Network, where Trap1 and Trap2 are the display names that appear as descriptors of this test in the monitor interface.
The test considers a configured OID for monitoring only when the actual value of the OID matches with its configured value. For instance, in the example above, if the value of OID .1.3.6.1.4.1.9156.1.1.2 is found to be HOST and not Host_system, then the test ignores OID .1.3.6.1.4.1.9156.1.1.2 while monitoring.
An * can be used in the OID/value patterns to denote any number of leading or trailing characters (as the case may be). For example, to monitor all the OIDs that return values which begin with the letter 'F', set this parameter to Failed:*-F*.
5. SHOWOID - Selecting the TRUE option against SHOWOID will ensure that the detailed diagnosis of this test shows the OID strings along with their corresponding values. If you select FALSE, then the values alone will appear in the detailed diagnosis page, and not the OIDs.
6. DETAILED DIAGNOSI - To make diagnosis more efficient and accurate, the eG Enterprise suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the On option. To disable the capability, click on the Off option.
The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled:
➢ The eG manager license should allow the detailed diagnosis capability
➢ Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.
</td></tr>
<tr><td>Outputs of the test</td><td>One set of results for every server being monitored</td></tr>
</table>

Measurements made by the test	Measurement	Measurement Unit	Interpretation
	SNMP traps received: Indicates the number of trap messages sent since the last measurement period.	Number	The detailed diagnosis of this measure, if enabled, provides the host from which an SNMP trap originated, the time at which the trap was sent, and the details of the trap.

1.12 Application Traps Test

The ApplicationTrap test reports the number of SNMP trap messages sent on account of errors in the transactions of various applications.

Purpose	Reports the number of SNMP trap messages sent on account of errors in the transactions of various applications
Target of the test	An SNMP trap
Agent deploying the test	An internal agent

<table>
<tr><td>Configurable parameters for the test</td><td>
1. TEST PERIOD - How often should the test be executed
2. HOST - The host for which the test is to be configured
3. PORT – The port at which the application listens
4. SOURCEADDRESS - Specify a comma-separated list of IP addresses or address patterns of the hosts sending the traps. For example, 10.0.0.1,192.168.10.*. A leading '*' signifies any number of leading characters, while a trailing '*' signifies any number of trailing characters.
5. OIDVALUE - Provide a comma-separated list of OID and value pairs returned by the traps. The values are to be expressed in the form, DisplayName:OID-OIDValue. For example, assume that the following OIDs are to be considered by this test: .1.3.6.1.4.1.9156.1.1.2 and .1.3.6.1.4.1.9156.1.1.3. The values of these OIDs are as given hereunder:
<table>
<tr><th>OID</th><th>Value</th></tr>
<tr><td>.1.3.6.1.4.1.9156.1.1.2</td><td>Host_system</td></tr>
<tr><td>.1.3.6.1.4.1.9156.1.1.3</td><td>NETWORK</td></tr>
</table>
In this case the OIDVALUE parameter can be configured as Trap1:.1.3.6.1.4.1.9156.1.1.2-Host_system,Trap2:.1.3.6.1.4.1.9156.1.1.3-Network, where Trap1 and Trap2 are the display names that appear as descriptors of this test in the monitor interface.

The test considers a configured OID for monitoring only when the actual value of the OID matches with its configured value. For instance, in the example above, if the value of OID .1.3.6.1.4.1.9156.1.1.2 is found to be HOST and not Host_system, then the test ignores OID .1.3.6.1.4.1.9156.1.1.2 while monitoring.

An * can be used in the OID/value patterns to denote any number of leading or trailing characters (as the case may be). For example, to monitor all the OIDs that return values which begin with the letter 'F', set this parameter to Failed:*-F*.
6. SHOWOID - Selecting the TRUE option against SHOWOID will ensure that the detailed diagnosis of this test shows the OID strings along with their corresponding values. If you select FALSE, then the values alone will appear in the detailed diagnosis page, and not the OIDs.
7. DETAILED DIAGNOSIS - To make diagnosis more efficient and accurate, the eG Enterprise suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the On option. To disable the capability, click on the Off option.

The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled:
➢ The eG manager license should allow the detailed diagnosis capability
➢ Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.
</td></tr>
<tr><td>Outputs of the test</td><td>One set of results for every server being monitored</td></tr>
</table>

Measurements made by the test	Measurement	Measurement Unit	Interpretation
	SNMP traps received: Indicates the number of trap messages sent since the last measurement period.	Number	The detailed diagnosis of this measure, if enabled, provides the host from which an SNMP trap originated, the time at which the trap was sent, and the details of the trap.

1.13 WebLogic Log Requests Test

The WebLogicLogRequests test monitors a web server access log and reports measures such as the number of requests that have been logged, the number of successful responses, the number of failed responses, etc., for every pattern that has been configured.

Purpose	Monitors a web server access log and reports measures such as the number of requests that have been logged, the number of successful responses, the number of failed responses, etc., for every pattern that has been configured
Target of the test	A WebLogic server
Agent deploying the test	An internal agent

Configurable parameters for the test	1. **TEST PERIOD** - How often should the test be executed 2. **HOST** - The host for which the test is to be configured 3. **PORT** – The port at which the server listens 4. **ABSOLUTEFILENAME** - Specify the full path to the log file to be monitored.. 5. **RECORDPATTERN** - The records in the log file that need to be considered for monitoring will have to be provided in the **RECORDPATTERN** text box. The pattern configuration should be in the following format: *{f0}sep1{f1}sep2{f2}*, where *{f0}*, *{f1}*, and *{f2}* represent the indexes of the first, second, and third fields (respectively) of the records logged in the log file, and *sep1* and *sep2* are the separators after {f0} and {f1} respectively. A separator can be a combination of any number of characters. For example, take the case of a log file with the following entry: *192.168.10.7 - - [12/Nov/1998:09:40:40 -0500] "POST /soap/servlet/helloworld HTTP/1.1" 200 3834* To ensure that the above record is considered for monitoring, the record pattern will have to be specified as follows: *{f0}- -{f1}"{f2}"{f3} {f4}*, where *{f0}* represents the first field of the record, which is followed by the separator '- -', and so on. 6. **SEARCHPATTERN** - Of the records that match the configured **RECORDPATTERN**, the eG agent will search for and monitor only those records which match the string patterns specified in the **SEARCHPATTERN** text box. To help you understand how to configure a **SEARCHPATTERN**, let us take the example of the following search pattern: *IP1:ALL,F0:192.168.10.7*,F3: 200*,COUNT(*),AVG(F4)*. ▪ Here, *IP1* is just a display name that will be displayed in the eG monitor interface as a descriptor of this test. ▪ The term *ALL* instructs the eG Enterprise system to consider only those records that fulfill all the conditions that follow. Alternatively, the key word *Any* can be used, which implies that the eG Enterprise system, while monitoring, will consider even those records that fulfill either of the conditions that follow. The conditions are: o *F0:192.168.10.7** indicates that for a record to be considered for monitoring, the first field (i.e. the field with index 0) of the record should begin with the IP *192.168.10.1*. Alternatively, the condition can be configured as *F0:192.168.10.7*+192.168.10.8*+192.168.10.9**, where '+' denotes an 'OR' operator. This configuration indicates that for a record to be considered for monitoring, the first field of the record should begin with any of the three values configured - i.e. 192.168.10.7, 192.168.10.8, or 192.168.10.9.

<table>
<tr><td></td><td colspan="3">

F3:200* indicates that for a record to be considered for monitoring, the fourth field (i.e. the field with index 3) of the record should begin with the number 200. Alternatively, the condition can be configured as F3:200*+300*+400*, where '+' denotes an 'OR' operator. This configuration indicates that for a record to be considered for monitoring, the fourth field of the record should begin with any of the three values configured - i.e. 200, 300, or 400.

COUNT(*) returns the number of records that fulfill the configured criteria.
AVG(F4) returns the average of the values of all the fields with index 4 (i.e. the fifth field), in the records that match the configured criteria.

According to this specification, the eG Enterprise system, while taking a count and while calculating the average, will consider only those records where the first field starts with '192.168.10.1' and the fourth field starts with '200'. The number '200' indicates a successful response. Therefore, this specification will report the metrics pertaining to only the successful responses for the IP patterns defined within the descriptor IP1 (i.e. 192.168.10.7*).

However, the test's configuration becomes complete only if the failure statistics are also extracted for IP1. Therefore, you will have to provide another search pattern for the descriptor IP1, so that the failure information is collected. The format of this pattern should be: IP1_FAIL: ALL,f0:192.168.10.7*,!f3:200*,COUNT(*),AVG(f4). Note that the descriptor names are the same, but the one meant for monitoring the failure cases, has been tagged as _FAIL. The specification !f3:200 indicates that the records with the number '200' (in the fourth field) should NOT be considered for monitoring. '!' is a NOT operator. Since '200' represents a success state, !200 ensures that only the failed responses for IP1 are considered for monitoring.

The complete SEARCHPATTERN will hence be: IP1:ALL,f0:192.168.10.7*,f3:200*,COUNT(*),AVG(f4)#& IP1_FAIL:ALL,f0:192.168.10.7*,!f3:200*,COUNT(*),AVG(f4), where #& is the separator.

In the monitor interface however, the descriptor IP1 alone will appear, but when clicked, will display both the success and failure statistics for the pattern 192.168.10.7*. Therefore, it is imperative that the WLLogReqTest be configured in such a way that it tracks both the success and failure cases for every IP pattern configured for monitoring. Otherwise, the test will not function as desired. This implies that if an IP pattern IP2 is configured for monitoring successful responses, then an IP2_FAIL should follow to monitor the failed responses. Similarly, multiple patterns can be configured for monitoring, separated by '#&'.
</td></tr>
<tr><td>Outputs of the test</td><td colspan="3">One set of results for every search pattern being configured</td></tr>
<tr><td rowspan="2">Measurements made by the test</td><td>Measurement</td><td>Measurement Unit</td><td>Interpretation</td></tr>
<tr><td>Total requests:
Indicates the number of account calls that are being made during a period of time.</td><td>Number</td><td>A high value of this measure indicates a heavy workload on the server.</td></tr>
</table>

	Successes: Indicates the number of successful responses.	Number	Low value of this measure indicates less number of successful responses from the server.
	Avg success bytes: Indicates the number of bytes of successful responses	Bytes	A high value of this measure indicates a high rate of successful responses.
	Failures: Indicates the number of failed responses.	Number	
	Avg fail bytes: Indicates the number of bytes of failed responses.	Bytes	A high value of this measure indicates a high failure rate.
	Avg bytes sent: Indicates the size (in bytes) of responses sent by the server.	Bytes	

Note:

If any of the measures of this test returns the value -5, then such a measure will not be displayed in the monitor interface. On the other hand, if all the measures of this test return the value -5, then all the measures will appear in the monitor interface, but the value displayed for each measure will be "Not Available".

1.14 WebLogic Log Responses Test

This test monitors an application log and reports measures such as the total number of responses that have been logged and average response time of every log file entry pattern that has been configured.

Purpose	Monitors an application log and reports measures such as the total number of responses that have been logged and average response time of every log file entry pattern that has been configured
Target of the test	A WebLogic server
Agent deploying the test	An internal agent

<table>
<tr><td>Configurable parameters for the test</td><td colspan="3">1. TEST PERIOD - How often should the test be executed
2. HOST - The host for which the test is to be configured
3. PORT – The port at which the server listens
4. ABSOLUTEFILENAME - Specify the full path to the log file to be monitored.
5. RECORDPATTERN - The records in the log file that need to be considered for monitoring will have to be provided in the RECORDPATTERN text box. The pattern configuration should be in the following format: {f0}sep1{f1}sep2{f2}, where {f0}, {f1}, and {f2} represent the indexes of the first, second, and third fields (respectively) of the records logged in the log file, and sep1 and sep2 are the separators after {f0} and {f1} respectively. A separator can be a combination of any number of characters.
For example, take the case of a log file with the following entries:
2486:Sampleappln:LoginUser->Time Taken for:LOGIN_CHECK; is:155
2530:Sampleappln:LoginUser->Time Taken for:AVAIL_CHECK; is:252
To ensure that the above records are considered for monitoring, the record pattern will have to be specified as follows: {f0}:{f1}:{f2}->{f3}:{f4}:{f5}, where {f0} represents the first field of the record, which is followed by the separator ':', and so on.
6. SEARCHPATTERN - Of the records that match the configured RECORDPATTERN, the eG agent will search for and monitor only those records which match the string patterns specified in the SEARCHPATTERN text box. To help you understand how to configure a SEARCHPATTERN, let us take the example of the following search pattern: Info1:ANY,f4:!LOGIN_CHECK*,COUNT(*),AVG(f5).
▪ Here, Info1 is just a display name that will be displayed in the eG monitor interface as a descriptor of this test.
▪ Use the term ALL or Any to instruct the eG Enterprise system to consider only those records that fulfill the condition that follows, for monitoring. The condition is: f4:!LOGIN_CHECK*. This indicates that for a record to be considered for monitoring, the fifth field (i.e. the field with index 4) of the record should 'not' begin with the string LOGIN_CHECK. The '!' symbol is the 'not' operator.
▪ COUNT(*) returns the number of records that fulfill the configured criteria.
▪ AVG(f5) returns the average of the values of all the fields with index 5 (i.e. the sixth field), in the records that match the configured criteria.
According to this specification, the eG Enterprise system, while taking a count and while calculating the average, will consider only those records where the fifth field does not begin with 'LOGIN_CHECK'. Similarly, multiple search patterns can be provided separated by "#&". For example, Info1:ANY,f4:!LOGIN_CHECK*,COUNT(*),AVG(f5)#&Info2:ALL,f4:AVAIL_CHECK *,COUNT(*),AVG(f5).</td></tr>
<tr><td>Outputs of the test</td><td colspan="3">One set of results for every server being monitored</td></tr>
<tr><td>Measurements made by the</td><td>Measurement</td><td>Measurement Unit</td><td>Interpretation</td></tr>
</table>

test	**Calls:** Indicates the number of account calls that are being made during a period of time.	Number	A high value of this measure indicates a heavy workload on the server.
	Avg response time: Indicates the average response time for account calls.	Secs	A dramatic increase in this value may be indicative of poor responsiveness of the server.
Note: If any of the measures of this test returns the value -5, then such a measure will not be displayed in the monitor interface. On the other hand, if all the measures of this test return the value -5, then all the measures will appear in the monitor interface, but the value displayed for each measure will be "Not Available".			

1.15 WebLogic Log Patterns Test

The WebLogicLogPatterns test monitors an application log and reports measures such as the total number of responses that have been logged and average response time of every log file entry pattern that has been configured.

Purpose	Monitors an application log and reports measures such as the total number of responses that have been logged and average response time of every log file entry pattern that has been configured
Target of the test	A WebLogic server
Agent deploying the test	An internal agent

<table>
<tr><td>Configurable parameters for the test</td><td colspan="3">1. TEST PERIOD - How often should the test be executed
2. HOST - The host for which the test is to be configured
3. PORT – The port at which the server listens
4. ABSOLUTEFILENAME - Specify the full path to the log file to be monitored.
5. RECORDPATTERN - The records in the log file that need to be considered for monitoring will have to be provided in the RECORDPATTERN text box. The pattern configuration should be in the following format: {f0}sep1{f1}sep2{f2}, where {f0}, {f1}, and {f2} represent the indexes of the first, second, and third fields (respectively) of the records logged in the log file, and sep1 and sep2 are the separators after {f0} and {f1} respectively. A separator can be a combination of any number of characters.
For example, take the case of a log file with the following entry:
eg_sample_appln_jsp :;TIME:2005-01-01 00:06:26.904;Thread_ID:ExecuteThread: '48' for queue: default';Duration:233
To ensure that the above record is considered for monitoring, the record pattern will have to be specified as follows: {f0};{f1};{f2};{f3}:{f4}, where {f0} represents the first field of the record, which is followed by the separator ';', and so on.
6. SEARCHPATTERN - Of the records that match the configured RECORDPATTERN, the eG agent will search for and monitor only those records which match the string patterns specified in the SEARCHPATTERN text box. To help you understand how to configure a SEARCHPATTERN, let us take the example of the following search pattern: Info1:any,f0:*eg_sample_appln_jsp *,count(*),avg(f4).
▪ Here, Info1 is just a display name that will be displayed in the eG monitor interface as a descriptor of this test.
▪ Use the term ALL or Any to instruct the eG Enterprise system to consider only those records that fulfill the condition that follows, for monitoring. The condition is: f0:*eg_sample_appln_jsp*. This indicates that for a record to be considered for monitoring, the first field (i.e. the field with index 0) of the record should embed the string eg_sample_appln_jsp.
▪ COUNT(*) returns the number of records that fulfill the configured criteria.
▪ AVG(f5) returns the average of the values of all the fields with index 5 (i.e. the sixth field), in the records that match the configured criteria.
According to this specification, the eG Enterprise system, while taking a count and while calculating the average, will consider only those records where the first field embeds the string eg_sample_appln_jsp. Similarly, multiple search patterns can be provided separated by "#&".</td></tr>
<tr><td>Outputs of the test</td><td colspan="3">One set of results for every server being monitored</td></tr>
<tr><td>Measurements made by the</td><td>Measurement</td><td>Measurement Unit</td><td>Interpretation</td></tr>
</table>

test	**Calls:** Indicates the number of account calls that are being made during a period of time.	Number	A high value of this measure indicates a heavy workload on the server.
	Avg response time: Indicates the average response time for account calls.	Secs	A dramatic increase in this value may be indicative of poor responsiveness of the server.

Note:

If any of the measures of this test returns the value -5, then such a measure will not be displayed in the monitor interface. On the other hand, if all the measures of this test return the value -5, then all the measures will appear in the monitor interface, but the value displayed for each measure will be "Not Available".

1.16 Large File Test

Some systems in a target environment could be hosting files of large sizes; a few of these files might not be of any use to either the user or the system (eg., *.tmp). In order to locate these files and remove them so as to conserve disk space, the LargeFileTest comes in handy. This test reveals the number of files in a specific directory that are of or above a configured size. If such large-sized files exist, then the detailed diagnosis of this test, when enabled, provides the names of the large files and their respective sizes.

Purpose	Reveals the number of files in a specific directory that are of or above a configured size
Target of the test	A host system
Agent deploying the test	An internal agent

<table>
<tr><td>Configurable parameters for the test</td><td colspan="3">1. TEST PERIOD - How often should the test be executed
2. HOST - The host for which the test is to be configured.
3. DIRECTORIES - Specify a comma-separated list of directories to be searched and file sizes, in the following format:{FULL_PATH_TO_DIR}@{FILE_SIZE}. For example, to check whether the directory c:\documents\important consists of files that are of size 2 MB or above, specify the following in the DIRECTORIES text box: c:\documents\important@2. Similarly, multiple {DIR}@{FILE_SIZE} combinations can be provided as a comma-separated list. For example: c:\documents\important@2,c:\letters\business@1. In case of Unix environments, this will be:/opt/docs@2,/opt/bin@3.
4. RECURSIVE - Set the RECURSIVE flag to yes to ensure that the test searches even the sub-directories within the configured DIRECTORIES for the files. By setting this flag to no, you can instruct the test to search for the files in the parent directory only.
5. DETAILED DIAGNOSIS - To make diagnosis more efficient and accurate, the eG Enterprise suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the On option. To disable the capability, click on the Off option.
The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled:
➢ The eG manager license should allow the detailed diagnosis capability
➢ Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.</td></tr>
<tr><td>Outputs of the test</td><td colspan="3">One set of results for every DIRECTORY configured</td></tr>
<tr><td rowspan="2">Measurements made by the test</td><td>Measurement</td><td>Measurement Unit</td><td>Interpretation</td></tr>
<tr><td>Largefiles count:
Indicates the number of files of or above a configured size in this directory.</td><td>Number</td><td>The detailed diagnosis of this test, if enabled, provides the names of the large files and their respective sizes.</td></tr>
</table>

1.17 SSL Certificate Test

All SSL web servers are configured with security certificates. During the SSL protocol handshake with clients, a server exchanges this certificate with the clients. An SSL certificate includes information about the server/domain to which the certificate is licensed, the issuing authority, and a validity period for the certificate. Beyond the validity period, the SSL certificate becomes invalid, and clients' SSL connections to the web server would fail. To avoid such a situation, it is essential that web server administrators are alerted in advance about the potential expiry of the SSL certificates on their web site. The SSLCertTest monitors the validity period for SSL certificates of different web sites.

Purpose	Monitors the validity period for SSL certificates of different web sites

Target of the test	A Web server		
Agent deploying the test	An internal agent		
Configurable parameters for the test	1. **TEST PERIOD** - How often should the test be executed 2. **HOST** - The host for which the test is to be configured. 3. **PORT** - The port at which the **HOST** listens 4. **TIMEOUT** - Provide the duration (in seconds) beyond which the test times out 5. **TARGETS** - Provide a comma-separated list of *{HostIP/Name}:{Port)* pairs, which represent the web sites to be monitored. For example, *192.168.10.7:443,192.168.10.8:443*. The test connects to each IP/port pair and checks for validity of the certificate associated with this target. One set of metrics is reported for each target. The descriptor represents the common name (CN) value of the SSL certificate		
Outputs of the test	One set of results for every **TARGET** configured		
Measurements made by the test	**Measurement**	**Measurement Unit**	**Interpretation**
	SSL certificate validity: Represents the validity of the SSL certificate in days.	Days	As this value approaches close to 0, an alert is generated to proactively inform the administrator that the SSL certificate is nearing expiry. A value of 0 indicates that the SSL certificate has expired.

1.18 Stratus Hardware Traps Test

This test monitors the status of various hardware elements present in the Stratus server using SNMP traps.

Purpose	Monitors the status of various hardware elements present in the Stratus server using SNMP traps
Target of the test	The Stratus server
Agent deploying the test	An internal agent

<table>
<tr><td>Configurable parameters for the test</td><td colspan="3">
1. TEST PERIOD - How often should the test be executed
2. HOST - Host name of the server for which the test is to be configured
3. PORT - The port at which the HOST listens
4. OIDVALUE - Provide a comma-separated list of OID and value pairs returned by the traps. The values are to be expressed in the form, DisplayName:OID-OIDValue. For example, assume that the following OIDs are to be considered by this test: .1.3.6.1.4.1.9156.1.1.2 and .1.3.6.1.4.1.9156.1.1.3. The values of these OIDs are as given hereunder:
<table>
<tr><th>OID</th><th>Value</th></tr>
<tr><td>.1.3.6.1.4.1.9156.1.1.2</td><td>Host_system</td></tr>
<tr><td>.1.3.6.1.4.1.9156.1.1.3</td><td>NETWORK</td></tr>
</table>
In this case the OIDVALUE parameter can be configured as Trap1:.1.3.6.1.4.1.9156.1.1.2-Host_system,Trap2:.1.3.6.1.4.1.9156.1.1.3-Network, where Trap1 and Trap2 are the display names that appear as descriptors of this test in the monitor interface.
The test considers a configured OID for monitoring only when the actual value of the OID matches with its configured value. For instance, in the example above, if the value of OID .1.3.6.1.4.1.9156.1.1.2 is found to be HOST and not Host_system, then the test ignores OID .1.3.6.1.4.1.9156.1.1.2 while monitoring.
An * can be used in the OID/value patterns to denote any number of leading or trailing characters (as the case may be). For example, to monitor all the OIDs that return values which begin with the letter 'F', set this parameter to Failed:*-F*.
5. SHOWOID - Selecting the TRUE option against SHOWOID will ensure that the detailed diagnosis of this test shows the OID strings along with their corresponding values. If you select FALSE, then the values alone will appear in the detailed diagnosis page, and not the OIDs.
6. DETAILED DIAGNOSIS - To make diagnosis more efficient and accurate, the eG Enterprise suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the On option. To disable the capability, click on the Off option.
The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled:
➢ The eG manager license should allow the detailed diagnosis capability
➢ Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.
</td></tr>
<tr><td>Outputs of the test</td><td colspan="3">One set of results for every OID value monitored</td></tr>
<tr><td>Measurements made by the test</td><td>Measurement</td><td>Measurement Unit</td><td>Interpretation</td></tr>
</table>

	Empty: Indicates that a slot in the system is in an "empty" state.	Boolean	For a slot, this state indicates that the slot is empty, physically not present, or electrically inaccessible. If the empty device causes the system to be go into simplex mode, the device is no longer fault tolerant. In some cases this state represents both a slot and a device. For instance, an instance of an SRA_DIMM in the Empty state means that a slot exists for the DIMM, but that the slot is empty. DIMMs, CPU Boards, IO Boards and Processors are represented by such WMI objects. Sensors go to this state instead of the"Not Present" state when they are not present. Empty devices are generally enumerable.
	Not present: Indicates that a device in the system is in a "not present" state.	Boolean	This state indicates that a device is either physically not present or electrically inaccessible. For instance, pulling the power cord on a CPU board makes the DIMMs and Processors on the board go to this state. When a WMI object goes to this state, it is generally not enumerable. Thus, this state only appears in state change events.
	Removed: Indicates that a device in the system is in a "removed" state. Usually, this is a final state but it can be a transient state.	Boolean	Usually, this state indicates that a device was intentionally removed from service. When intentionally removed from service, the device remains in this state. Only some devices go to this state when removed from services; other devices go to other offline states. Some devices pass through this state as they are brought online.
	Dumping: Indicates that a device is in a "Dumping" state. This is a transient state.	Boolean	This state indicates a device is in the process of writing a dump to a file.
	Diagnostics passed: Indicates that a device is in a "Diagnostic Passed" state. This is a transient state and the device should change to "online" state when it is brought online.	Boolean	This state indicates that a device has just completed its diagnostics tests.

	Initialising: Indicates that a device is in a "Initialising" state. This is a transient state and the device should change to "online" state when it is brought online.	Boolean	This state indicates that a device is in the process of initializing.
	Syncing: Indicates that a device is in a "synching" state. This is a transient state and the device should change to "online" state when it is brought online.	Boolean	This state indicates that a device is synchronizing itself with its partners. For instance, when a CPU is brought up, it synchronizes its memory and its processor state with that of its partners.
	Offline: Indicates that a device is in a "offline" state.	Boolean	This state indicates that a device is offline. Only some devices can go to this state while other devices go into the "Removed From Service" state.
	Firmware update complete: Indicates that a device's firmware update procedure has completed.	Boolean	
	Diagnostics: Indicates that a device is running diagnostics.	Boolean	
	Online: Indicates that a device is in a "online" state.	Boolean	This state indicates that the device is online, but not configured for redundancy. For instance, a working NIC that is not part of a team will be in this state. Although the online state does not indicate whether a device is safe-to-pull or not, on a properly configured system such devices can be assumed safe-to-pull.

	Simplex: Indicates that a device is in a "Simplex" state.	Boolean	This state indicates that a device is online, configured for redundancy, and is not safe-to-pull. When applied to a port, indicates that the port is configured for redundancy, and that whatever is connected to the port is not safe-to-pull.
	Duplex: Indicates that a device is in a "Duplex" state.	Boolean	This state indicates that a device is online, configured for redundancy, and is safe-to-pull. When applied to a port, indicates that the port is configured for redundancy, and that whatever is connected to the port is safe-to-pull.
	Shot: Indicates that a device is in a "Shot" state. This is a transient state and the device should either transit to "broken" or "online" state after diagnostic is done.	Boolean	This state indicates that a device experienced a problem and will soon move to either an online state or the broken state.
	Broken: Indicates that a device is in a "Broken" state.	Boolean	This state Indicates that a device is broken. In the case of a port, this state may mean that the port is inoperative or that that which attaches to the port is inoperative. There are several reasons that a device could be broken but usually points to hardware errors. Contact your service providers for service checks. In the case where the device is a port, it usually indicates that there is nothing attached to the port, or when whatever should be attached to the port is not responding. For example, a NIC port will be in this state when it cannot detect link.

1.19 Process Activity Test

The ProcessActivity test reports statistics related to the number and size of processes executing on a system. This test works on Solaris, Linux, HPUX, and AIX platforms only.

Purpose	Reports statistics related to the number and size of processes executing on a system
Target of the test	Solaris, Linux, AIX, and HPUX systems
Agent	An internal agent

<table>
<tr><td>deploying the test</td><td colspan="3"></td></tr>
<tr><td>Configurable parameters for the test</td><td colspan="3">1. TEST PERIOD - How often should the test be executed
2. HOST - The host for which the test is to be configured
3. PORT – The port at which the HOST listens
4. PROCESS - Enter a comma separated list of processNames:processPattern pairs which identify the process(es) executing on the server under consideration. processName is a string that will be used for display purposes only. processPattern is an expression of the form - *expr* or expr or *expr or expr* or *expr1*expr2*... or expr1*expr2, etc. A leading '*' signifies any number of leading characters, while a trailing '*' signifies any number of trailing characters. For example, the PROCESS parameter can contain the following value: Java:*java*. Here, Java is the pattern name that will be displayed in the eG monitor interface as the info (descriptor) of the ProcActivityTest. The Java pattern in our example will monitor those processes, the names of which embed the string 'java'.
5. DETAILED DIAGNOSIS - To make diagnosis more efficient and accurate, the eG Enterprise suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the On option. To disable the capability, click on the Off option.
The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled:
➢ The eG manager license should allow the detailed diagnosis capability
➢ Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.</td></tr>
<tr><td>Outputs of the test</td><td colspan="3">One set of results for the every process pattern configured</td></tr>
<tr><td rowspan="3">Measurements made by the test</td><td>Measurement</td><td>Measurement Unit</td><td>Interpretation</td></tr>
<tr><td>Current processes:
Indicates the number of processes currently running.</td><td>Number</td><td></td></tr>
<tr><td>Processes added:
Indicates the number of processes added during the last measurement period.</td><td>Number</td><td></td></tr>
</table>

	Processes removed: Indicates the number of processes that were abnormally terminated/completed during the last measurement period.	Number	
	Virtual size: Indicates the total size of the process in virtual memory.	MB	
	Resident size: Indicates the resident size of the process. This denotes the size taken up by the process in the RAM, i.e., real address space.	MB	Virtual size is always greater than or equal to the resident size of the process. This measure will not be available for AIX platforms.

1.20 SQL Response Test

The responsiveness of a database to SQL queries is not only indicative of the health of the database server, but also the efficiency of the queries. A well-tuned database is one that quickly responds to SQL queries, and a well-built SQL query is one that succeeds in retrieving the desired results from the database and that too, in record time. The SQLResponseTest monitors SQL queries from start to finish, and reports the status of the query execution and its responsiveness. This way, administrators are proactively notified of failed queries and queries that take too long to execute, so that root-cause diagnosis is instantly inititated.

Purpose	Monitors SQL queries from start to finish, and reports the status of the query execution and its responsiveness
Target of the test	A database server
Agent deploying the test	An internal agent

Configurable parameters for the test

1. **TEST PERIOD** - How often should the test be executed
2. **HOST** - The host for which the test is to be configured
3. **PORT** – The port at which the **HOST** listens
4. **JDBC_DRIVER** - Specify the JDBC driver that is used to access the database. The table below lists the JDBC drivers that correspond to some of the most popular database servers that are monitored by eG Enterprise. Refer to this table whenever in need.

Database	Driver
Oracle	oracle.jdbc.driver.OracleDriver
MS SQL	net.sourceforge.jtds.jdbc.Driver
Informix	com.informix.jdbc.IfxDriver
Sybase	com.sybase.jdbc2.jdbc.SybDriver
MySql	org.gjt.mm.mysql.Driver

5. **CONNECTION_URL** - Specify the JDBC URL for the database. The URL format is JDBC driver specific. The table below lists the JDBC URLs for some of the most popular database servers that are monitored by eG Enterprise. While configuring this test for any of the database servers in this table, you can specify a URL of the corresponding format.

Database	URL Format
Oracle	jdbc:oracle:thin:@**{host}:{port}:{instance}**
MS SQL	jdbc:jtds:sqlserver://**{host}:{port}/{database}**
Informix	jdbc:informix-sqli://**{host}:{port}/{database}:informixserver={instance}**
Sybase	jdbc:sybase:Tds:**{host}:{port}/{database}**
MySql	jdbc:mysql://**{host}:{port}/{database}**

If the target database is not in the above list, then follow the steps given below:

- Download the JDBC driver of the new database from the database vendor.
- Copy the relevant java package files (jar or zip) into the **{EG_AGENT_INSTALL_DIR}\lib** directory (on Windows; on Unix, this will be the **opt/egurkha/lib** directory).
- If a Unix agent is executing this test, then simply proceed to restart the eG agent. In case of a Windows agent however, edit the **debugoff.bat** file in the **{EG_AGENT_INSTALL_DIR}\lib** directory to manually set the classpath value. Then, execute **debugoff.bat** so that the agent service is reinstalled on Windows with the new classpath settings.

	➢ Next, login to the eG administrative interface and configure this test with the **JDBC_DRIVER** and **CONNECTION_URL** that corresponds to the new database. 6. **USER** - Te name of the **USER** who is vested with the privilege to execute the configured query. 7. **PASSWORD** - The password of the **USER**. 8. **CONFIRM PASSWORD** - Confirm the password by retyping it in the **CONFIRM PASSWORD** text box. 9. **QUERY** - specify the query to be executed and monitored.		
Outputs of the test	One set of results for the database server monitored		
Measurements made by the test	**Measurement**	**Measurement Unit**	**Interpretation**
	Query status: Indicates whether the configured query has been successfully executed.	Boolean	The value of 1 indicates successful execution, and 0 indicates failure. In case of query failure, you can use the detailed diagnosis of this measure, if enabled, to view the errors that caused the query to fail; troubleshooting thus becomes easier.
	Query time: Indicates the time taken to execute the query and retrieve results.	Secs	An abnormally high value is a cause for concern, and warrants further investigation.

1.21 Memory Status - NetSnmp

This test provides memory statistics by polling the NetSNMP MIB.

Purpose	Provides memory statistics by polling the NetSNMP MIB
Target of the test	
Agent deploying the test	External/Remote agent

Configurable parameters for the test	1. **TEST PERIOD** - How often should the test be executed 2. **HOST** – The IP address of the Cisco Router. 3. **SNMPPORT** - The port number through which the device exposes its SNMP MIB. The default value is 161. 4. **SNMPVERSION** – By default, the eG agent supports SNMP version 1. Accordingly, the default selection in the **SNMPVERSION** list is **v1**. However, if a different SNMP framework is in use in your environment, say SNMP **v2** or **v3**, then select the corresponding option from this list. 5. **SNMPCOMMUNITY** – The SNMP community name that the test uses to communicate with the Cisco router. This parameter is specific to SNMP **v1** and **v2** only. Therefore, if the **SNMPVERSION** chosen is **v3**, then this parameter will not appear. 6. **USERNAME** – This parameter appears only when **v3** is selected as the **SNMPVERSION**. SNMP version 3 (SNMPv3) is an extensible SNMP Framework which supplements the SNMPv2 Framework, by additionally supporting message security, access control, and remote SNMP configuration capabilities. To extract performance statistics from the MIB using the highly secure SNMP v3 protocol, the eG agent has to be configured with the required access privileges – in other words, the eG agent should connect to the MIB using the credentials of a user with access permissions to be MIB. Therefore, specify the name of such a user against the **USERNAME** parameter. 7. **AUTHPASS** – Specify the password that corresponds to the above-mentioned **USERNAME**. This parameter once again appears only if the **SNMPVERSION** selected is **v3**. 8. **CONFIRM PASSWORD** – Confirm the **AUTHPASS** by retyping it here. 9. **AUTHTYPE** – This parameter too appears only if **v3** is selected as the **SNMPVERSION**. From the **AUTHTYPE** list box, choose the authentication algorithm using which SNMP v3 converts the specified **USERNAME** and **PASSWORD** into a 32-bit format to ensure security of SNMP transactions. You can choose between the following options: ➢ **MD5** – Message Digest Algorithm ➢ **SHA** – Secure Hash Algorithm 10. **ENCRYPTFLAG** – This flag appears only when **v3** is selected as the **SNMPVERSION**. By default, the eG agent does not encrypt SNMP requests. Accordingly, the **ENCRYPTFLAG** is set to **NO** by default. To ensure that SNMP requests sent by the eG agent are encrypted, select the **YES** option. 11. **ENCRYPTTYPE** – If the **ENCRYPTFLAG** is set to **YES**, then you will have to mention the encryption type by selecting an option from the **ENCRYPTTYPE** list. SNMP v3 supports the following encryption types: ➢ **DES** – Data Encryption Standard ➢ **AES** – Advanced Encryption Standard 12. **ENCRYPTPASSWORD** – Specify the encryption password here. 13. **CONFIRM PASSWORD** – Confirm the encryption password by retyping it here.

	14. **TIMEOUT** - Specify the duration (in seconds) within which the SNMP query executed by this test should time out in the **TIMEOUT** text box. The default is 10 seconds.		
Outputs of the test	One set of results for every router being monitored.		
Measurements made by the test	**Measurement**	**Measurement Unit**	**Interpretation**
	Total swap: Indicates the total amount of swap space configured for this host.	MB	
	Available swap: Indicates the amount of swap space currently unused or available.	MB	
	Swap availability: Indicates the percentage of the unused or available swap memory.	Percent	A very low value indicates that the swap space configured may not be sufficient. A value close to 100% may imply that the swap space configured may be too large.
	Real memory: Indicates the total amount of real/physical memory installed on this host.	MB	
	Available real memory: Indicates the amount of real/physical memory currently unused or available.	MB	
	Free memory: Indicates the total amount of memory free or available for use on this host.	MB	A very low value of free memory is also an indication of high memory utilization on a host.
	Shared memory: Indicates the total amount of real or virtual memory currently allocated for use as shared memory.	MB	

	Buffer memory: Indicates the total amount of real or virtual memory currently allocated for use as memory buffers.	MB	
	Cached memory: Indicates the total amount of real or virtual memory currently allocated for use as cached memory.	MB	

1.22 Disk Status - NetSnmp

This test provides disk usage statistics by polling the NetSNMP MIB.

Purpose	Provides disk usage statistics by polling the NetSNMP MIB
Target of the test	
Agent deploying the test	External/Remote agent

Configurable parameters for the test	1. **TEST PERIOD** - How often should the test be executed 2. **HOST** – The IP address of the Cisco Router. 3. **SNMPPORT** - The port number through which the device exposes its SNMP MIB. The default value is 161. 4. **SNMPVERSION** – By default, the eG agent supports SNMP version 1. Accordingly, the default selection in the **SNMPVERSION** list is **v1**. However, if a different SNMP framework is in use in your environment, say SNMP **v2** or **v3**, then select the corresponding option from this list. 5. **SNMPCOMMUNITY** – The SNMP community name that the test uses to communicate with the Cisco router. This parameter is specific to SNMP **v1** and **v2** only. Therefore, if the **SNMPVERSION** chosen is **v3**, then this parameter will not appear. 6. **USERNAME** – This parameter appears only when **v3** is selected as the **SNMPVERSION**. SNMP version 3 (SNMPv3) is an extensible SNMP Framework which supplements the SNMPv2 Framework, by additionally supporting message security, access control, and remote SNMP configuration capabilities. To extract performance statistics from the MIB using the highly secure SNMP v3 protocol, the eG agent has to be configured with the required access privileges – in other words, the eG agent should connect to the MIB using the credentials of a user with access permissions to be MIB. Therefore, specify the name of such a user against the **USERNAME** parameter. 7. **AUTHPASS** – Specify the password that corresponds to the above-mentioned **USERNAME**. This parameter once again appears only if the **SNMPVERSION** selected is **v3**. 8. **CONFIRM PASSWORD** – Confirm the **AUTHPASS** by retyping it here. 9. **AUTHTYPE** – This parameter too appears only if **v3** is selected as the **SNMPVERSION**. From the **AUTHTYPE** list box, choose the authentication algorithm using which SNMP v3 converts the specified **USERNAME** and **PASSWORD** into a 32-bit format to ensure security of SNMP transactions. You can choose between the following options: ➢ **MD5** – Message Digest Algorithm ➢ **SHA** – Secure Hash Algorithm 10. **ENCRYPTFLAG** – This flag appears only when **v3** is selected as the **SNMPVERSION**. By default, the eG agent does not encrypt SNMP requests. Accordingly, the **ENCRYPTFLAG** is set to **NO** by default. To ensure that SNMP requests sent by the eG agent are encrypted, select the **YES** option. 11. **ENCRYPTTYPE** – If the **ENCRYPTFLAG** is set to **YES**, then you will have to mention the encryption type by selecting an option from the **ENCRYPTTYPE** list. SNMP v3 supports the following encryption types: ➢ **DES** – Data Encryption Standard ➢ **AES** – Advanced Encryption Standard 12. **ENCRYPTPASSWORD** – Specify the encryption password here. 13. **CONFIRM PASSWORD** – Confirm the encryption password by retyping it here.

	14. **TIMEOUT** - Specify the duration (in seconds) within which the SNMP query executed by this test should time out in the **TIMEOUT** text box. The default is 10 seconds.		
Outputs of the test	One set of results for every router being monitored.		
Measurements made by the test	**Measurement**	**Measurement Unit**	**Interpretation**
	Total size: Indicates the total size of each disk/partition.	MB	
	Free space: Indicates the available space on the disk.	MB	Ideally, the value of this measure should be high.
	Used space: Indicates the used space on the disk.	MB	
	Percent usage: Indicates the percentage of space used on disk.	Percent	A value close to 100% is a cause for concern, as it indicates that the disk is running out of space.
	Inodes used: Indicates the percentage of inodes used on disk.	Percent	

1.23 CPU Status - NetSnmp

This test provides CPU usage statistics by polling the NetSNMP MIB.

Purpose	Provides CPU usage statistics by polling the NetSNMP MIB
Target of the test	
Agent deploying the test	External/Remote agent

Configurable parameters for the test	1. **TEST PERIOD** - How often should the test be executed 2. **HOST** – The IP address of the Cisco Router. 3. **SNMPPORT** - The port number through which the device exposes its SNMP MIB. The default value is 161. 4. **SNMPVERSION** – By default, the eG agent supports SNMP version 1. Accordingly, the default selection in the **SNMPVERSION** list is **v1**. However, if a different SNMP framework is in use in your environment, say SNMP **v2** or **v3**, then select the corresponding option from this list. 5. **SNMPCOMMUNITY** – The SNMP community name that the test uses to communicate with the Cisco router. This parameter is specific to SNMP **v1** and **v2** only. Therefore, if the **SNMPVERSION** chosen is **v3**, then this parameter will not appear. 6. **USERNAME** – This parameter appears only when **v3** is selected as the **SNMPVERSION**. SNMP version 3 (SNMPv3) is an extensible SNMP Framework which supplements the SNMPv2 Framework, by additionally supporting message security, access control, and remote SNMP configuration capabilities. To extract performance statistics from the MIB using the highly secure SNMP v3 protocol, the eG agent has to be configured with the required access privileges – in other words, the eG agent should connect to the MIB using the credentials of a user with access permissions to be MIB. Therefore, specify the name of such a user against the **USERNAME** parameter. 7. **AUTHPASS** – Specify the password that corresponds to the above-mentioned **USERNAME**. This parameter once again appears only if the **SNMPVERSION** selected is **v3**. 8. **CONFIRM PASSWORD** – Confirm the **AUTHPASS** by retyping it here. 9. **AUTHTYPE** – This parameter too appears only if **v3** is selected as the **SNMPVERSION**. From the **AUTHTYPE** list box, choose the authentication algorithm using which SNMP v3 converts the specified **USERNAME** and **PASSWORD** into a 32-bit format to ensure security of SNMP transactions. You can choose between the following options: ➢ **MD5** – Message Digest Algorithm ➢ **SHA** – Secure Hash Algorithm 10. **ENCRYPTFLAG** – This flag appears only when **v3** is selected as the **SNMPVERSION**. By default, the eG agent does not encrypt SNMP requests. Accordingly, the **ENCRYPTFLAG** is set to **NO** by default. To ensure that SNMP requests sent by the eG agent are encrypted, select the **YES** option. 11. **ENCRYPTTYPE** – If the **ENCRYPTFLAG** is set to **YES**, then you will have to mention the encryption type by selecting an option from the **ENCRYPTTYPE** list. SNMP v3 supports the following encryption types: ➢ **DES** – Data Encryption Standard ➢ **AES** – Advanced Encryption Standard 12. **ENCRYPTPASSWORD** – Specify the encryption password here. 13. **CONFIRM PASSWORD** – Confirm the encryption password by retyping it here.

	14. **TIMEOUT** - Specify the duration (in seconds) within which the SNMP query executed by this test should time out in the **TIMEOUT** text box. The default is 10 seconds.		
Outputs of the test	One set of results for every router being monitored.		
Measurements made by the test	**Measurement**	**Measurement Unit**	**Interpretation**
	Total CPU usage: Indicates the total CPU usage of the server.	Percent	A high value could signify a CPU bottleneck. The CPU utilization may be high because a few processes are consuming a lot of CPU, or because there are too many processes contending for a limited resource. Check the currently running processes to see the exact cause of the problem.
	User CPU: Indicates the percentage of CPU that is being used for user processes.	Percent	An unusually high value indicates a problem and may be due to too many user tasks executing simultaneously.
	System CPU: Indicates the percentage of CPU that is being used for system processes.	Percent	An unusually high value indicates a problem and may be due to too many system-level tasks executing simultaneously.
	Nice CPU: Indicates the percentage of CPU being used by Nice processes (i.e., processes that do not have the default priority).	Percent	
	Idle CPU: Indicates the percentage of time that the server is idle.	Percent	

1.24 Directory Updates Test

This test monitors specific directories for files that are older than a configured duration.

Purpose	Monitors specific directories for files that are older than a configured duration
Target of the test	
Agent deploying the	Internal agent

test	

Configurable parameters for the test	1. **TEST PERIOD** - How often should the test be executed 2. **HOST** – The IP address of the host. 3. **PORT** – The port at which the **HOST** listens. 4. **DIRECTORY_LIST** – This text box takes a comma seperated list of directory paths that are to be monitored. For example, if you want to monitor a directory called *temp* in the C drive, then you need to specify, *c:\temp*. If you would like to monitor a directory named *root* which is a sub-directory of *temp*, then your specification should be: *c:\temp\root*. To monitor both the *temp* and *root* directories in our example, specify the following in the **DIRECTORY_LIST** text box: *c:\temp,c:\temp\root*.. 5. **HOURS_OLDER** – This test reports the number of old files in the configured directories. In the **HOURS_OLDER** text box therefore, you need to specify how old the files in the specified directory have to be, so that they are considered for monitoring by this test. For example, if the **DIRECTORY_LIST** contains *c:\temp*, and the **HOURS_OLDER** text box contains the value 2, then the test will report the number of files in the *temp* directory that were last modified over (i.e., greater than) 2 hours before. For every directory specification in the **DIRECTORY_LIST**, you can specify a corresponding value in the **HOURS_OLDER** text box - i.e., if 3 directories are configured in the **DIRECTORY_LIST**, then the **HOURS_OLDER** can also contain a comma-separated list of 3 values - say, *2,3,4*. In this case, the test will report the following: o For the first directory in the **DIRECTORY_LIST**, the test will report the number of files in the directory that were last modified over 2 hours ago. o For the second directory in the **DIRECTORY_LIST**, the test will report the number of files in the directory that were last modified over 3 hours before. o For the third directory in the **DIRECTORY_LIST**, the test will report the number of files in the directory that were last modified over 4 hours ago. Alternatively, you can also specify a single value in the **HOURS_OLDER** text box. This value will automatically apply to all the directories configured in the **DIRECTORY_LIST**. In other words, **the number of values that you specify in the HOURS_OLDER text box should either be 1 or should be equal to the number of directories configured in the DIRECTORY_LIST**. 6. **DETAILED DIAGNOSIS** - To make diagnosis more efficient and accurate, the eG Enterprise suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the **On** option. To disable the capability, click on the **Off** option. The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled: ➢ The eG manager license should allow the detailed diagnosis capability ➢ Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.

Outputs of the test	One set of results for every directory in the **DIRECTORY_LIST**		
Measurements made by the test	Measurement	Measurement Unit	Interpretation
	Number of old files: Indicates the number of old files in this directory.	Number	In the event that the host runs out of space, you might want to check the value of this measure to figure out if there are too many old files. If so, then you can use the detailed diagnosis of this test to identify the old files, determine whether you still need the files, and if found useless, remove the files so as to make space in the directory.

1.25 Windows Memory Stats Test

This test reports details about the physical memory of the system.

Purpose	Reports details about the physical memory of the system		
Target of the test	A Windows host		
Agent deploying the test	Internal agent		
Configurable parameters for the test	1. **TEST PERIOD** - How often should the test be executed 2. **HOST** – The IP address of the host. 3. **PORT** – The port at which the **HOST** listens.		
Outputs of the test	One set of results for the host being monitored		
Measurements made by the test	Measurement	Measurement Unit	Interpretation
	Committed memory in use: Indicates the committed bytes as a percentage of the Commit Limit.	Percent	In the event that the host runs out of space, you might want to check the value of this measure to figure out if there are too many old files. If so, then you can use the detailed diagnosis of this test to identify the old files, determine whether you still need the files, and if found useless, remove the files so as to make space in the directory.

	Pool nonpaged failures: Indicates the number of times allocations have failed from non paged pool.	Number	Generally, a non-zero value indicates a shortage of physical memory.
	Pool paged failures: Indicates the number of times allocations have failed from paged pool.	Number	A non-zero value indicates a shortage of physical memory.
	Copy read hits: Indicates the percentage of copy read calls satisfied by reads from the Cache out of all read calls.	Percent	Any value over 80% is excellent.

1.26 Windows Interrupts Test

This test reports how busy the system processor was while handling hardware device interrupts.

Purpose	Reports how busy the system processor was while handling hardware device interrupts
Target of the test	A Windows host
Agent deploying the test	Internal agent
Configurable parameters for the test	1. **TEST PERIOD** - How often should the test be executed 2. **HOST** – The IP address of the host. 3. **PORT** – The port at which the **HOST** listens.
Outputs of the test	One set of results for the host being monitored

Measurements made by the test	Measurement	Measurement Unit	Interpretation
	Interrupt time: Indicates the percentage of time spent by the processor for receiving and servicing the hardware interrupts during the last polling interval.	Percent	This is an indirect indicator of the activity of devices that generate interrupts such as system Clocks, the mouse device drivers, data communication lines, network interface cards and other peripheral devices. In general, a very high value of this measure might indicate that a disk or network adapter needs upgrading or replacing.