Monitoring eG Agent

eG Innovations Product Documentation

www.eginnovations.com

eG
Total Performance Visibility

Table of Contents

Table of Figures

Chapter 1: Introduction

The purpose of an IT infrastructure monitoring solution is to observe the performance of the applications, devices, and systems that underlie their mission-critical service offerings, proactively detect potential deviations, and promptly alert administrators to the root-cause of service outages/slowdowns. To be able to provide such critical performance monitoring and fault management services in a timely, reliable, and uninterrupted manner, the monitoring solution , like the components it monitors, should perform at peak capacity at all times!

An efficient monitoring solution is one that not only monitors other applications but also analyzes its own performance at periodic intervals. This way, the solution can instantly capture snags in its operations, accurately point administrators to the source of these performance degradations, and facilitate the speedy resolution of issues. In the absence of such self-monitoring capabilities, you may end up in a situation where slowdowns experienced by the monitoring solution - say, owing to a overload condition, or agent is unable to collect metrics or is unable to report to the manager, causing issues to go unnoticed or to aggravate.

To avoid such unpleasant eventualities, the eG Enterprise Suite provides a dedicated eG Agent monitoring model. To know more about monitoring the eG Agent refer to, **Monitoring the eG Agent**.

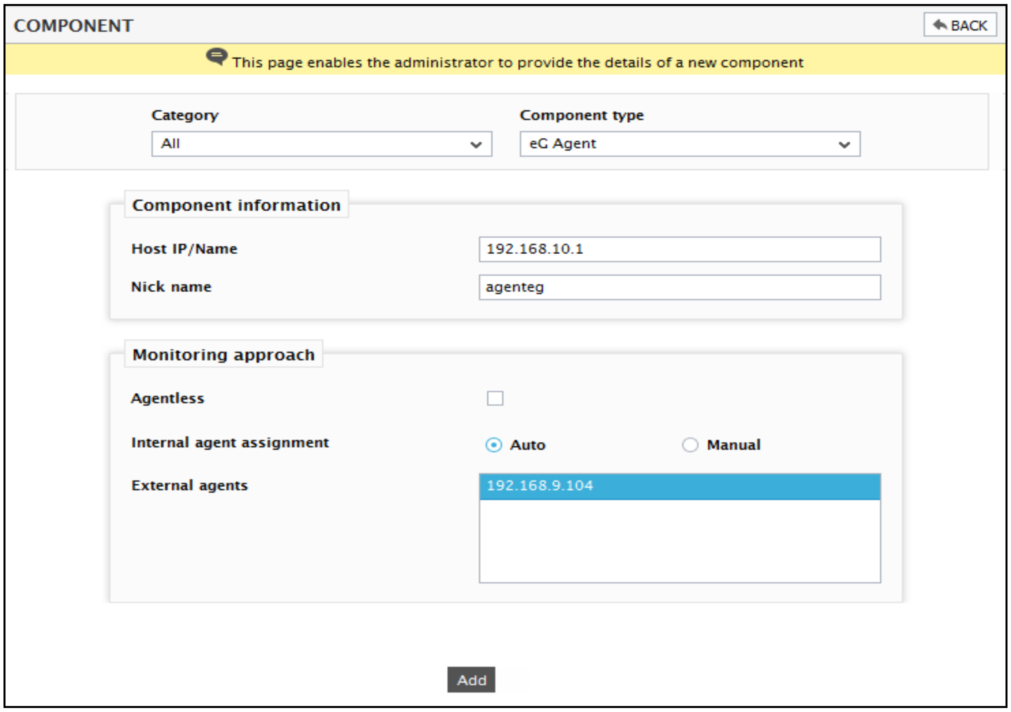
Chapter 2: How to Monitor eG Agent Using eG Enterprise?

eG Enterprise monitors the eG agent in an agent-based manner.

2.1 Managing the eG Agent

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

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



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

Category: All

Component type: eG Agent

Component information

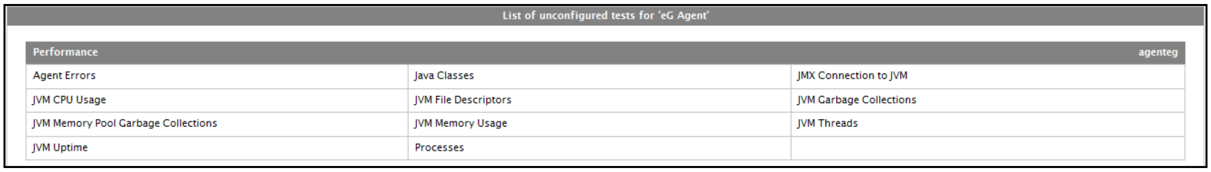
Host IP/Name	192.168.10.1
Nick name	agenteg

Monitoring approach

Agentless	□
Internal agent assignment	● Auto ○ Manual
External agents	192.168.9.104

Figure 2.1: Adding an eG Agent component

4. Specify **Host IP/name** and **Nick name** for the eG Agent component as shown in Figure 2.1. Then, click the **Add** button to save the changes.
5. When you attempt to sign out, a list of unconfigured tests will appear as shown in Figure 2.2.



List of unconfigured tests for 'eG Agent'

Performance		agenteg
Agent Errors	Java Classes	JMX Connection to JVM
JVM CPU Usage	JVM File Descriptors	JVM Garbage Collections
JVM Memory Pool Garbage Collections	JVM Memory Usage	JVM Threads
JVM Uptime	Processes	

Figure 2.2: List of unconfigured tests to be configured for the eG Agent

6. For more information on configuring the tests refer to **Monitoring the eG Agent** chapter.
7. Once all the tests are configured, sign out of the eG administrative interface.

Chapter 3: Monitoring the eG Agent

The eG Enterprise Suite provides a dedicated eG Agent monitoring model that monitors the functioning of the eG agent at configured frequencies and reports abnormalities (if any).

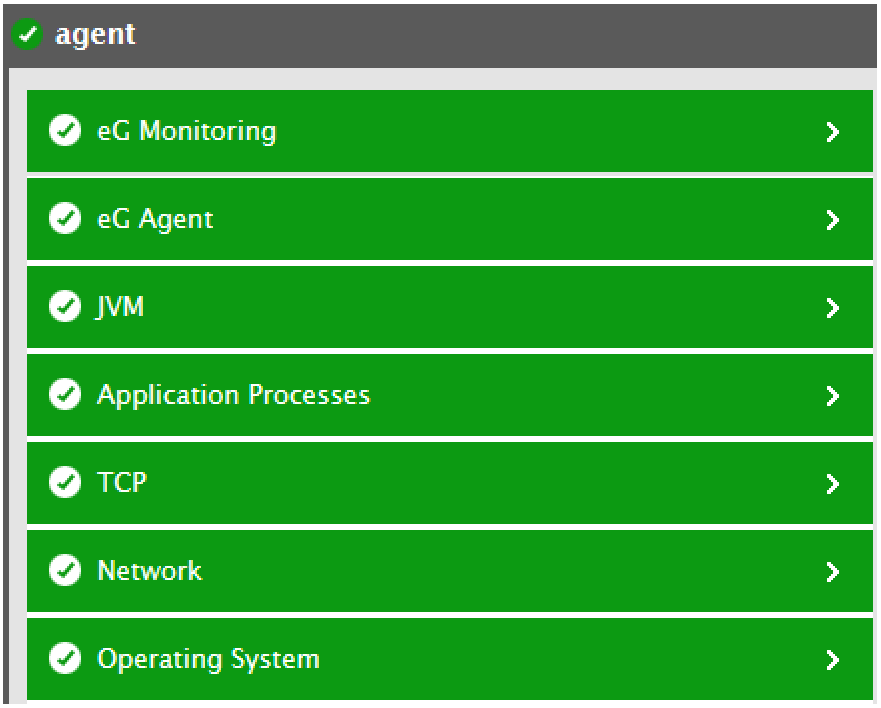


Figure 3.1: Layer model of the eG Agent

Each layer of Figure 1 above is mapped to tests that report a wealth of performance information related to the eG agent. To pull out these performance metrics, the eG agent can be deployed on the eG agent host itself (agent-based) or on any remote Windows host in the environment (agentless).

By executing the tests pertaining to the eG agent, eG administrators can find quick and accurate answers for the following queries:

- Is the eG agent communicating with the eG manager?
- For how many components did each test execute and report metrics to the eG agent?
- What is the minimum time and the maximum time taken to execute each test?
- What is the average time taken to execute each test?
- How many tester threads are available for executing each test?
- How many tester threads are currently executing each test?

- What is the percentage of tester threads utilized by each test?
- How well each test is reporting metrics in terms of percentage?
- How many tests are executing and reporting metrics to the target eG agent?
- How many files were stored in the data folder of the target eG agent?
- What is the size of the files stored in the data folder of the target eG agent?
- How long does the eG agent take to upload the data collected by executing the tests to the eG manager?
- How many virtual machines/desktops are monitored by the target eG agent?
- How many virtual machines/desktops are currently powered on?
- From how many virtual machines/desktops did the target eG agent collect the inside view metrics successfully?
- How many virtual machines/desktops from which the target eG agent was unable to collect the inside view metrics?

The sections that follow will discuss the top 2 layers of Figure 3.1 only, as the bottom 4 layers have been discussed extensively in the *Monitoring Unix and Windows Servers* document and the **JVM** layer has been discussed in the *Monitoring Java Applications* document.

3.1 The eG Agent Layer

Using this layer, administrators can detect the agent manager communication, possible overload condition of the eG agent and the count of the errors logged in the error log file of the eG agent.

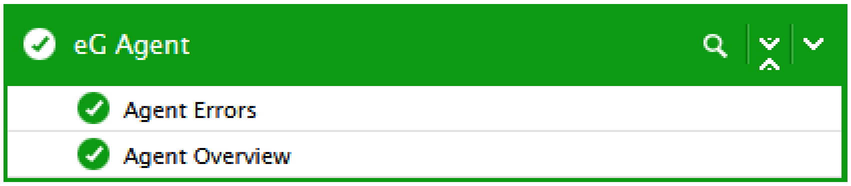


Figure 3.2: The tests pertaining to the eG Agent layer

Let us now discuss each test of this layer in detail in the forthcoming sections.

3.1.1 Agent Errors Test

This test monitors the error log file of the target eG agent and reports the number of errors of specific error patterns available in the error log file.

Target of the test : An eG Agent

Agent deploying the test : An internal agent

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

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The host for which the test is being configured.
Port	Specify the port at which the specified host listens in the Port text box. By default, this is *NULL*.
AlertFile	specify the path to the alert log file to be monitored. For eg., */user/john/alert_john.log*. Multiple log file paths can be provided as a comma-separated list - e.g., */user/john/alert_egurkha.log,/tmp/log/alert.log*. 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. 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 monitored during the last measurement period; • Whether any new log files (that match the AlertFile specification) have been newly added since the last measurement period; 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).
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.

Parameter	Description
	"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 the log file path, 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>:**.
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 as the value for Lines and if the corresponding value in the SearchPattern field is like Error:ERROR:*, Fail:*failed, then: 0:0 will be applied to Error:ERROR:* pattern 1:1 will be applied to Fail:*failedpattern
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'*.
UniqueMatch	By default, this parameter is set to **False**, indicating that, by default, the test checks every line in the log file for the existence of each of the configured SearchPatterns. By setting this parameter to **True**, you can instruct the test to ignore a line and move to the next as soon as a match for one of the configured patterns is found in that line. For example, assume that *Error:*ERROR:*, Fail:*failed** is the SearchPattern that has been configured. If UniqueMatch is set to **False**, then the test will read every line in the log file completely to check for the existence of messages embedding the strings 'ERROR:' and 'failed'. If both the patterns are detected in the same line, then the

Parameter	Description
	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 'ERROR:' and 'failed' follow one another in the same line, the test will consider only the first match and not the next. The match count in this case will therefore be incremented by only 1.
RotatingFile	This flag governs the display of descriptors for this test in the eG monitoring console. If this flag is set to **True**, 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.
CaseSensitive	This flag is set to **No** by default. This indicates that the test functions in a 'case-insensitive' manner by default. This implies that, by default, the test ignores the case of your AlertFile and SearchPattern specifications. If this flag is set to **Yes** on the other hand, then the test will function in a 'case-sensitive' manner. In this case therefore, for the test to work, even the case of your AlertFile and SearchPattern specifications should match with the actuals.
RolloverFile	By default, this flag is set to **False**. Set this flag to **True** if you want the test to support the 'roll over' capability (if any) of the specified AlertFile. A roll over typically occurs when the timestamp of a file changes or when the log file size crosses a pre-determined threshold. When a log file rolls over, the errors/warnings that pre-exist in that file will be automatically copied to a new file, and all errors/warnings that are captured subsequently will be logged in the original/old file.In such a scenario, since the RolloverFile flag is set to **False** by default, the test by default scans only the original/old file for new log entries and ignores the new file. On the other hand, if the flag is set to **True**, then the test will scan both old and the rolled-over file for new entries. 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 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.

Parameter	Description
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 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.
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. Where multiple log files are being monitored, you will have to provide a comma-separated list of encoding formats – one each for every log file monitored - eg., UTF-8, UTF-16. Make sure that your encoding format specification follows the same sequence as your AlertFile specification. In other words, the first encoding format should apply to the first alert file, and so on. 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.,D:\logs\report.log,E:\logs\error.log.03*15, D:\logs\report.log,E:\logs\error.log.04*15), then such configurations will only be supported in the ANSI format, and not the UTF format.
UseUTF8	If UTF-8 encoding is to be used for reading the specified log file, then, set the UseUTF8 flag to **true**. By default, this flag is set to **false**. If multiple log files are being monitored, then, for each file, you will have to indicate whether UTF-8 encoding is to be used for reading that file or not. For instance, assume that the AlertFile parameter is set to *dblogs@/tmp/db/dblogs.log,applogs@/tmp/app/applogs.log*. Now, to instruct the test to use UTF-8 encoding for reading the 'dblogs' log file and not to use the UTF-8 encoding while reading the 'applogs' log file, your USEUTF8 setting should be as follows: *true,false*. **Note that the number of values provided against the UseUTF8 parameter should be equal to the number of log files being monitored. Also, note that if the AlertFile being monitored has BOM, then the test will automatically use UTF-8 encoding to read that file, even if the UseUTF8 flag is set to false.** **Note:**

Parameter	Description
	If your AlertFile specification consists of file patterns that include wildcard characters (eg., */tmp/db/*dblogs*,/tmp/app/*applogs**), then the files that match such patterns will only support the ANSI format, and not the UTF format, even if the UTF-8 parameter is set to **true** for such patterns.
UseUTF16	If UTF-16 encoding is to be used for reading the specified log file, then, set the UseUTF16 flag to true. By default, this flag is set to false. If multiple log files are being monitored, then, for each file, you will have to indicate whether UTF-16 encoding is to be used for reading that file or not. For instance, assume that the AlertFile parameter is set to *dblogs@/tmp/db/dblogs.log,applogs@/tmp/app/applogs.log*. Now, to instruct the test to use UTF-16 encoding for reading the 'dblogs' log file and not to use the UTF-16 encoding while reading the ‘applogs’ log file, your UseUTF8 setting should be as follows: *true,false*. **Note that the number of values provided against the UseUTF16 parameter should be equal to the number of log files being monitored.** **Note:** If your AertFile specification consists of file patterns that include wildcard characters (eg., */tmp/db/*dblogs*,/tmp/app/*applogs**), then the files that match such patterns will only support the ANSI format, and not the UTF format, even if the UTF-16 parameter is set to **true** for such patterns.
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.
Detailed Diagnosis	To make diagnosis more efficient and accurate, the eG Enterprise suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the **On** option. To disable the capability, click on the **Off** option. The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled: • The eG manager license should allow the detailed diagnosis capability • Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Recent errors	Indicates the number of errors that were added to the error log file when the test was last executed.	Number	The value of this measure is a clear indicator of the number of “new” alerts that were logged into the error log of the monitored eG agent.

3.1.2 Agent Overview Test

To receive precise, real-time updates on the performance and overall health of the components being monitored, administrators need to ensure that the eG agent, which performs the actual monitoring functions, is communicating with the eG manager at all times. The **Agent Overview** test helps administrators in this regard.

This test periodically monitors the communication between the eG agent and the eG manager and also reports administrators on the numerical statistics of the tests that were executing for the eG agent, the files stored in the data folder and the size of the files stored in the data folder. This test also helps administrators to identify the time taken to upload the data from the eG agent to the eG manager.

Target of the test : The eG Agent

Agent deploying the test : An internal agent

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

Configurable parameters for the test

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

Parameter	Description
	DD frequency.
Detailed Diagnosis	To make diagnosis more efficient and accurate, the eG Enterprise suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the **On** option. To disable the capability, click on the **Off** option. The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled: • The eG manager license should allow the detailed diagnosis capability • Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Total running tests	Indicates the total number of tests that were executed by the eG agent.	Number	
Data storage files	Indicates the total number of files that were stored in the data folder of the eG agent.	Number	
Data storage file size	Indicates the amount of space utilized by the files available in the data folder of the eG agent.	MB	A high value for this measure for a considerable period of time indicates that the the eG agent could not communicate with the eG manager.
Is agent communicating with the manager?	Indicates whether/not the eG agent is communicating with the eG Manager.		The values reported by this measure and its numeric equivalents are mentioned in the table below:

<table>
<tr><th>Measurement</th><th>Description</th><th>Measurement Unit</th><th>Interpretation</th></tr>
<tr><td></td><td></td><td></td><td><table><tr><th>Measure Value</th><th>Numeric Value</th></tr><tr><td>Yes</td><td>1</td></tr><tr><td>No</td><td>0</td></tr></table>
<b>Note:</b>
By default, this measure reports the <b>Measure Value</b>s listed in the table above to indicate whether/not the eG agent is communicating with the eG Manager. However, the graph of this measure is represented using the numeric equivalents only i.e., 0 or 1.
The detailed diagnosis of this measure lists the last communication time of the eG agent and the eG Manager.</td></tr>
<tr><td>Time taken for upload</td><td>Indicates the total time taken by the eG agent to upload the data collected by executing the tests to the eG manager.</td><td>Seconds</td><td></td></tr>
</table>

3.1.3 Tests Test

This test auto-discovers the tests that report the data to the target eG agent and reports the number of components for which each test was executed by the eG agent and the time duration (maximum/minimum/average) taken for test execution. In addition, this test throws the spotlight on the tester threads that are available for starting the execution of each test. Using this test, administrators can easily figure out how well the tester threads were utilized for executing each test.

Target of the test : The eG Agent

Agent deploying the test : An internal agent

Outputs of the test : One set of results for each Test from which metrics are collected by the target eG Agent being monitored.

Configurable parameters for the test

Parameter	Description
Test period	How often should the test be executed .
Host	The host for which the test is to be configured.
Port	The port number at which the specified host listens. By default, this is *NULL*.
Detailed Diagnosis	To make diagnosis more efficient and accurate, the eG Enterprise suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the **On** option. To disable the capability, click on the **Off** option. The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled: • The eG manager license should allow the detailed diagnosis capability • Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Total components managed by this agent	Indicates the total number of components for which this test was executed and metrics collected from this test by the target eG agent.	Number	
Minimum frequency	Indicates the minimum time taken to execute this test.	Mins	The value of this measure could be between 4.5 to 5.5 minutes. A value higher/lower than the specified limit is a cause of concern.
Maximum frequency	Indicates the maximum time taken to execute this test.	Mins	The value of this measure could be between 4.5 to 5.5 minutes. A value higher/lower than the specified limit is a cause of concern.

Measurement	Description	Measurement Unit	Interpretation
			The detailed diagnosis of this measure lists the components for which this test executes with the maximum time duration.
Average frequency	Indicates the average time taken to execute this test.	Mins	
Test reporting ratio	Indicates how well the test was performing i.e., how well the test was reporting metrics to the target eG agent, in terms of percentage.	Percentage	A value greater than 100 for this measure indicates that the test was not executing in the configured measurement period. It may also indicate that the test took too long to execute.
Tester busy	Indicates the percentage of time the tester threads configured for this test were busy.	Percentage	Tester threads on the eG agent are responsible for starting the execution of the test and collecting appropriate metrics. A low value for this measure indicates that more components can be managed and monitored by the eG agent. A value close to 100 is a cause of concern.
Total testers	Indicates the total number of tester threads that were currently available for executing this test.	Number	
Running testers	Indicates the total number of tester threads that were currently executing this test.	Number	
Tester Utilization	Indicates the percentage of tester threads utilized by this test.	Percentage	A low value is desired for this measure. If the value of this measure is close to 100%, it could indicate that the test is about to utilize its entire thread pool. Under such circumstances, you can consider increasing the maximum number of threads that the test can spawn for execution.

3.2 The eG Monitoring Layer

Using this layer, administrators can determine the number of tests reporting to the eG agent , the time taken for executing the tests and the percentage utilization of the tester threads. In addition, administrators can also determine the number of VMs monitored by the eG agent and the number of VMs from which the eG agent could collect inside view metrics.

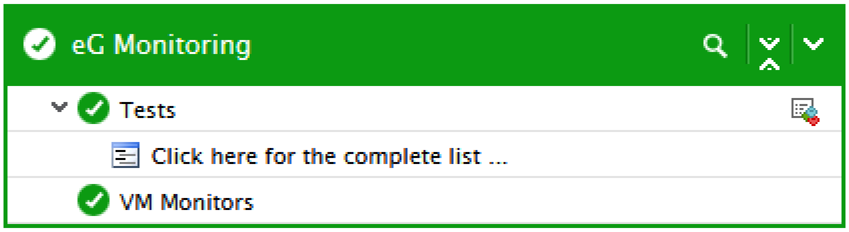


Figure 3.3: The tests pertaining to the eG Monitoring layer

Let us now discuss each test of this layer in detail in the forthcoming sections.

3.2.1 VM Monitors Test

To receive precise, real time updates on the virtual machines that are monitored by the target eG agent, administrators can rely on the **VM Monitors** test. This test reports the total number of virtual machines/desktops from which measures are collected by the target eG agent. In addition, this test reports the number of powered on virtual machines/desktops and provides the numerical statistics on the virtual machines/desktops on which the target eG agent sent separate internal probes and collected metrics pertaining to each VM. This way, this test helps administrators to figure out the VMs from which metrics could not be collected by the eG agent.

Target of the test : The eG Agent

Agent deploying the test : An internal agent

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

Configurable parameters for the test

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

Parameter	Description
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.
Detailed Diagnosis	To make diagnosis more efficient and accurate, the eG Enterprise suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the **On** option. To disable the capability, click on the **Off** option. The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled: • The eG manager license should allow the detailed diagnosis capability • Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Total VMs monitored	Indicates the total number of VMs from which metrics are collected by the target eG agent.	Number	The detailed diagnosis of this measure lists the name of the VMs for which metrics are collected by the eG agent.
Powered on VMs monitored	Indicates the number of powered on VMs from which the metrics are collected by the eG agent.	Number	
Successfully authenticated VMs	Indicates the number of VMs from which the eG agent was able to collect the *inside view* metrics ie., the metrics pertaining to each VM.	Number	
Authentication failed	Indicates the number of	Number	A high value for this measure indicates

Measurement	Description	Measurement Unit	Interpretation
VMs	VMs from which the *inside view* metrics could not be collected by the target eG agent.		that the inside view monitoring capability of the eG agent is failing. The detailed diagnosis of this measure lists the name of the Physical server on which the VMs are hosted, the IP address of the VMs and the reason behind the eG agent not being able to collect the internal metrics from the VMs.

About eG Innovations

eG Innovations provides intelligent performance management solutions that automate and dramatically accelerate the discovery, diagnosis, and resolution of IT performance issues in on-premises, cloud and hybrid environments. Where traditional monitoring tools often fail to provide insight into the performance drivers of business services and user experience, eG Innovations provides total performance visibility across every layer and every tier of the IT infrastructure that supports the business service chain. From desktops to applications, from servers to network and storage, from virtualization to cloud, eG Innovations helps companies proactively discover, instantly diagnose, and rapidly resolve even the most challenging performance and user experience issues.

eG Innovations is dedicated to helping businesses across the globe transform IT service delivery into a competitive advantage and a center for productivity, growth and profit. Many of the world's largest businesses use eG Enterprise to enhance IT service performance, increase operational efficiency, ensure IT effectiveness and deliver on the ROI promise of transformational IT investments across physical, virtual and cloud environments.

To learn more visit www.eginnovations.com.

Contact Us

For support queries, email support@eginnovations.com.

To contact eG Innovations sales team, email sales@eginnovations.com.