Monitoring SAP TREX Server

eG Innovations Product Documentation

www.eginnovations.com

Total Performance Visibility

Table of Contents

CHAPTER 1: INTRODUCTION 1

CHAPTER 2: MONITORING THE SAP TREX SERVER 2

2.1 Pre-Requisites for monitoring SAP TREX server 3

2.2 The Application Processes Layer 3

2.2.1 Trex Alerts Test 4

2.3 The Trex Server Layer 11

2.3.1 Global Dictionary Fill Level Test 12

2.3.2 External Location Indices Consistency Test 13

2.3.3 Fast Search Infrastructure Indices Check Test 15

2.3.4 Trex Indices Test 17

2.3.5 Trex Queues Test 21

2.3.6 Trex Threads Test 26

2.4 The Trex Services Layer 29

2.4.1 RFC Configuration Test 30

2.4.2 Trex Web Availability Test 33

2.4.3 Trex WorkLoad Test 34

ABOUT EG INNOVATIONS 40

Table of Figures

Figure 1.1: The SAP TREX architecture 1

Figure 2.1: The layer model of the SAP TREX server 2

Figure 2.2: The tests mapped to the Application Processes layer 4

Figure 2.3: The tests mapped to the Trex Server layer 12

Figure 2.4: The tests mapped to the Trex Services layer 30

Chapter 1: Introduction

TREX is based on a client/server architecture. The client component is integrated into the application that uses the TREX functions, and allows communication with the TREX servers. The server component processes the requests; it indexes and classifies documents and answers search queries. The server component comprises of the following individual servers:

- Web server with TREX extension
- RFC server
- Queue server
- Preprocessor
- Index server
- Name server

The figure below illustrates the individual components and the communication between components:

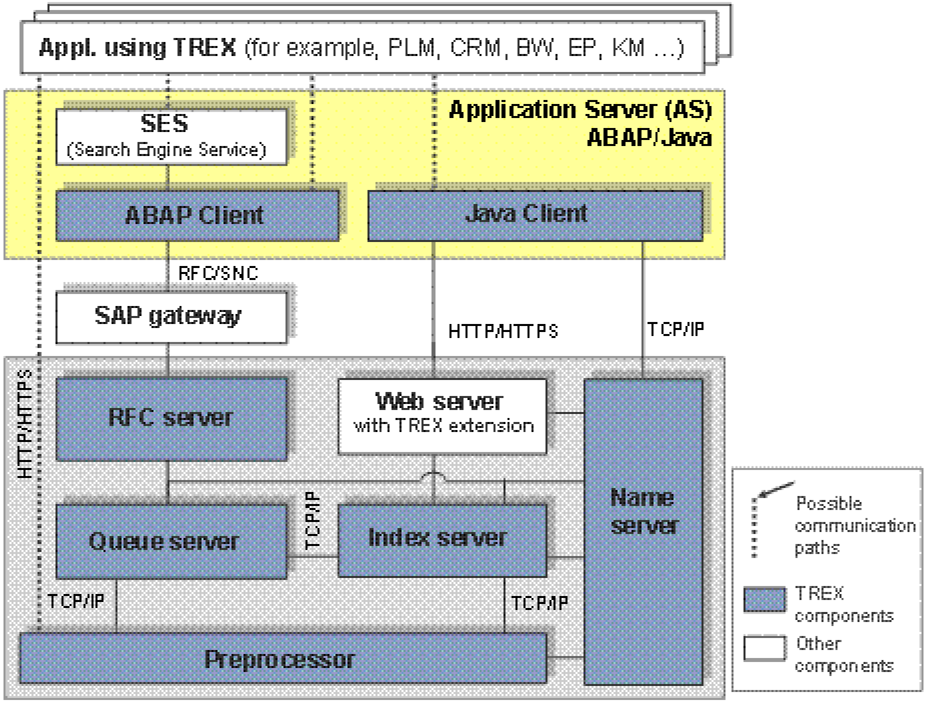


Figure 1.1: The SAP TREX architecture

Chapter 2: Monitoring the SAP TREX server

The eG Enterprise suite embeds a specialized monitoring model for the SAP TREX server (see Figure 2.1), using which the performance of the critical services and components of the server can be tracked, issues affecting server-performance captured at the earliest, and the root-cause of the issues promptly traced and treated before it adversely impacts the business in the overall SAP environment.

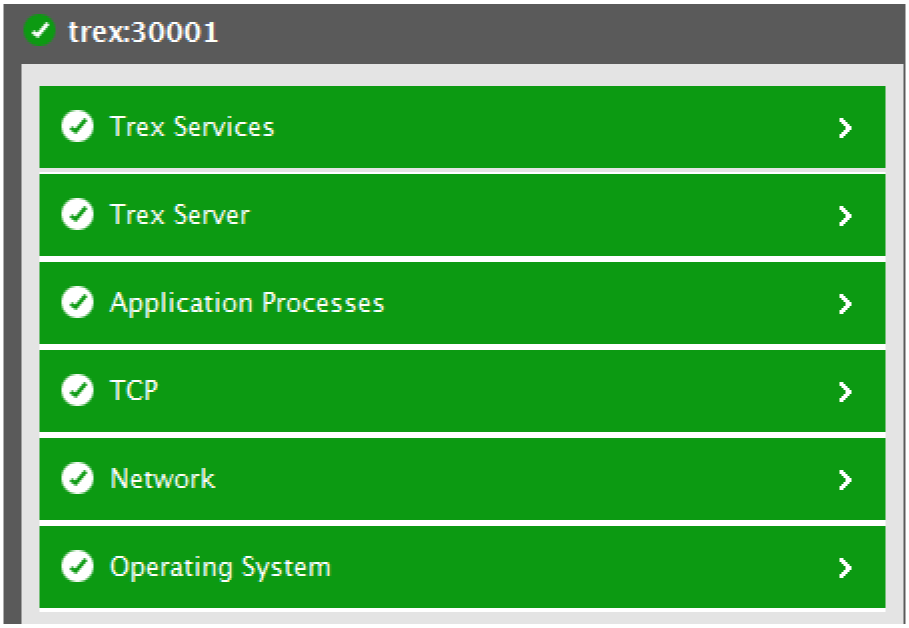


Figure 2.1: The layer model of the SAP TREX server

Every layer depicted by Figure 2.1 is associated with a series of tests, each of which seeks to answer the following questions related to the performance of the SAP TREX server:

- Is the server accessible when a HTTP request is simulated?;
- What is the current configuration status of the RFC server?;
- How many threads were configured for the RFC server?;
- What is the current status of the RFC server?;
- What is the current CPU and disk usage of the SAP TREX server?;
- How well the search requests, indexing requests and merging requests were processed by the SAP TREX server?;
- How many indices were available in a location other than the default directory?;

- How many indices were valid?;
- How many indexes were valid in the Fast Search infrastructure module?;
- What is the maximum size allocated to the global dictionary and what is the current size of the global dictionary?;
- What is the current status of each index?;
- How effective was each index in search and query operations?;
- What is the current state of the queues within each queue group?;
- What is the current state of documents in the queues that are within each queue group?;
- How many threads are allocated to each service of the SAP TREX server?;
- How many threads are currently running, idle, suspended etc for each service?;
- How many messages of the configured patterns were added to the logs?

Since the **Operating System**, **TCP** and **Network** layers of Figure 2.1 have already been discussed in detail in the *Monitoring Unix and Windows Servers* document, let us now focus on the remaining layers in the forthcoming sections.

2.1 Pre-Requisites for monitoring SAP TREX server

The eG Enterprise Suite is capable of monitoring the SAP TREX server in an *agent-based* manner. Prior to monitoring the SAP TREX server, administrators should ensure that the eG agent installed user has the rights to execute the scripts on the directory in which the SAP TREX server has been installed.

2.2 The Application Processes Layer

For a SAP TREX server instance, this layer is mapped to a *Trex Alerts* test, which will quickly capture errors/warnings encountered by the SAP TREX server. Apart from this test, this layer is mapped to the *Processes* test which reports on the status and resource usage of process patterns that have been explicitly configured for monitoring. In addition to these tests, a *SAP Process Status* test also runs on this layer which is elaborately discussed in the *Monitoring the SAP ABAP Instance* document. This layer is also mapped to a host of other tests that are discussed already in the *Monitoring Unix and Windows Servers* document.

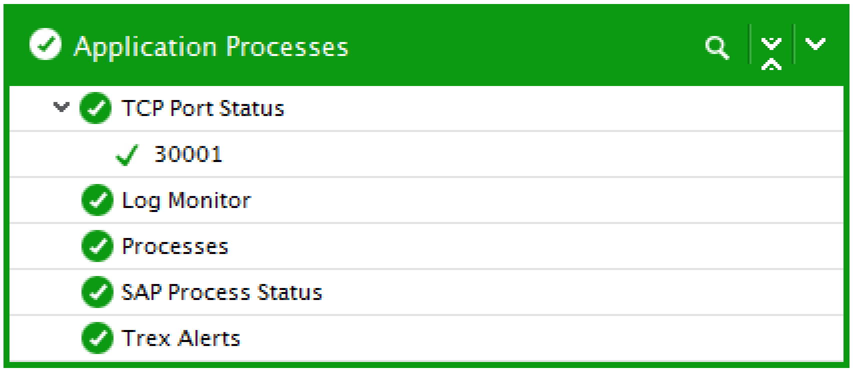


Figure 2.2: The tests mapped to the Application Processes layer

2.2.1 Trex Alerts Test

To enable administrators to quickly capture errors/warnings encountered by the SAP TREX server, administrators can use the **Trex Alerts** test.

This test scans the logs for errors/warnings of configured patterns and reports the number of entries in the log that match the configured patterns. Detailed metrics provided by the test also provides detailed message descriptions, so as to ease troubleshooting and hasten problem resolution.

Target of the test : A SAP TREX

Agent deploying the test : An internal agent

Outputs of the 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 -** Specify the port at which the specified **HOST** listens.
4. **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 detailed 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 error:error-*,offline:*offline*,online:*online then:

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

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

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

5. **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:\trex\logs\alert.txt*, and **ROTATINGFILE** is set to **true**, then, your descriptor will be of the following format: *c:\trex\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:\trex\logs*, and **ROTATINGFILE** is set to **true**, then, your descriptor will be: *c:\trex\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:\trex\logs\*alert**, and **ROTATINGFILE** is set to **true**, then, your descriptor will be: **alert*:<SearchPattern>*. In this case, the descriptor format will not change even if the **ROTATINGFILE** flag status is changed.

6. **LOGFILEPATH -** This test extracts the performance metrics from the diagnostics.log file present in the {ITS_INSTALL_DIR}\6.20\{Directory corresponding to the ITS instance}\logs directory. Therefore, in the LOGFILEPATH text box, provide the full path to the diagnostics.log file in the following format: {Instance Name}={Path to the log file}. For example, if the log file for an instance named 'ADM' is to be monitored, and ITS is installed in the C:\Program Files\SAP\ITS directory, then the LOGFILEPATH specification should be as follows: ADM=c:\Progra~1\SAP\ITS\6.20\ADM\logs\diagnostics.log. To monitor the diagnostics.log files associated with multiple instances, provide the LOGFILEPATH as a comma-separated list. For example, ADM=c:\Progra~1\SAP\ITS\6.20\ADM\logs\diagnostics.log,ITS1=c:\Progra~1\SAP\ITS\6.20\ITS1\logs\diagnostics.log.

7. **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 - eg., /user/john/alert_egurkha.log,/tmp/log/alert.log.

   Also, instead of a specific log file path, the path to the directory containing log files can be provided - e.g., */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.

   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).

8. **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.

9. **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**.

10. **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 your **SEARCHPATTERN** specification is as follows: DEBUG:DEBUG*. This indicates that *"DEBUG"* is the pattern name to be displayed in the monitor interface. "DEBUG*" indicates that the test will monitor only those lines in the specified log file which start with the string " *DEBUG* ". Similarly, if your pattern specification reads: ERROR:*ERROR, then it means that the pattern name is *ERROR* and that the test will monitor those lines in the log which end with the term *ERROR*.

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: DEBUG:*DEBUG*,ERROR:*ERROR*,INFO:*INFO*,ERROR:*ERROR*,WARNING:*WARN*,FATAL:*FATAL*

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>:**.

11. **USEUTF8 -** If UTF-8 encoding is to be used for reading the specified log file, then, set this 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:

    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.

12. **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 **true**. 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 **USEUTF16** 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.**

    Note:

    If your **ALERTFILE** specification consists of file patterns that include wildcard characters (e.g., */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.

13. **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'.

14. **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.

15. **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.

16. **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*.

17. **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**.

18. **USE SUDO -** The **USE SUDO** parameter is applicable to Unix environments only. By default, the eG agent does not require any special permissions to parse and read messages from the log file to be monitored. This is why, the **USE SUDO** parameter is set to No by default. In some highly-secure Unix environments however, the eG agent install user may not have the permission to read the log file to be monitored. In such environments, you will have to follow the steps below to ensure that the test is able to read the log file and report metrics:

    - Edit the SUDOERS file on the target host and append an entry of the following format to it:

      <eG_agent_install_user> ALL=(ALL) NOPASSWD: <Log_file_with_path>;

      For instance, if the eG agent install user is eguser, and the log file to be monitored is /usr/bin/logs/procs.log, then the entry in the SUDOERS file should be:

      eguser ALL=(ALL) NOPASSWD: /usr/bin/logs/procs.log

    - Finally, save the file.
    - Then, when configuring this test using the eG admin interface, set the **USE SUDO** parameter to Yes. Once this is done, then every time the test runs, it will check whether the eG agent install user has the necessary permissions to read the log file. If the user does not have the permissions, then the test runs the sudo command to change the permissions of the user, so that the eG agent is able to read from the log file.

19. **SUDO PATH -** The **SUDO PATH** parameter is relevant only when the **USE SUDO** parameter is set to *'Yes'*. By default, the **SUDO PATH** is set to *none*. This implies that the sudo command is in its default location - i.e., in the */usr/bin* or */usr/sbin* folder of the target host. In this case, once the **USE SUDO** flag is set to **Yes** , the eG agent automatically runs the sudo command from its default location to allow access to the configured log file. However, if the sudo command is available in a different location in your environment, you will have to explicitly specify the full path to the sudo command in the **SUDO PATH** text box to enable the eG agent to run the sudo command.

20. **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.

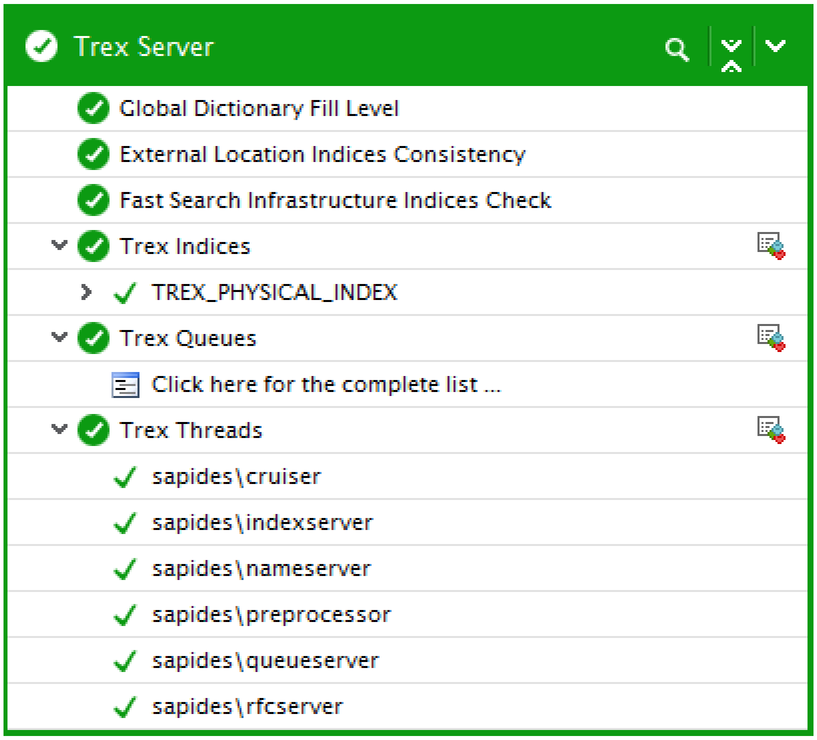
Measures made by the test:

Measurement	Description	Measurement Unit	Interpretation
New messages:	Indicates the number of messages of the configured patterns that were added to the logs 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 monitored logs. The detailed diagnosis of this measure if enabled, lists the recent messages.

2.3 The Trex Server Layer

This layer helps administrators to figure out the following:

- The number of indices that were available in a location other than the default directory;
- The number of indices that were valid;
- The number of valid indexes in the Fast Search infrastructure module;
- The maximum size allocated to the global dictionary and the current size of the global dictionary.
- The current status of each index;
- How effective was each index in search and query operations?
- What is the current state of the queues within each queue group?
- What is the current state of documents in the queues that are within each queue group?
- How many threads are allocated to each service of the SAP TREX server?
- How many threads are currently running, idle, suspended etc for each service?



Trex Server

- Global Dictionary Fill Level
- External Location Indices Consistency
- Fast Search Infrastructure Indices Check
- Trex Indices
  - TREX_PHYSICAL_INDEX
- Trex Queues
  - Click here for the complete list ...
- Trex Threads
  - sapides\cruiser
  - sapides\indexserver
  - sapides\nameserver
  - sapides\preprocessor
  - sapides\queueserver
  - sapides\rfcserver

Figure 2.3: The tests mapped to the Trex Server layer

2.3.1 Global Dictionary Fill Level Test

Dictionaries are used for compressed indexes for accelerator technologies like BI accelerator. A Global directory stores the data in the physical tables that translate values into index IDs. This directory would assign consecutive ID numbers, since renumbering the complete global dictionary below a newly inserted value would be infeasible considering the huge size of the dictionary. However, the dictionary is sorted to allow searches in logarithmic time and to perform range searches. If the global dictionary is near full, then, data search would take too long to complete which may result in performance bottleneck. Therefore, it is necessary to constantly keep a vigil on the size and fill level of the Global dictionary. This is where the Global Dictionary Fill Level test helps!

This test reports the maximum size allocated to the global dictionary for each index server and the current size of the global dictionary.

Target of the test : A SAP TREX

Agent deploying the test : An internal agent

Outputs of the test : One set of results for each *index server: Global dictionary* of the target SAP TREX server that is to be monitored

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 -** Specify the port at which the specified **HOST** listens.
4. **TREX DIRECTORY -** Specify the full path to the SAP TREX instance that is to be monitored for e.g., the location can be specified as *D:\usr\sap\TRX\TRX01*.
5. **TIMEOUT -** Specify the duration (in seconds) beyond which this test should time out in the **TIMEOUT** text box. The default is 120 seconds.

Measures made by the test:

Measurement	Description	Measurement Unit	Interpretation
Current size:	Indicates the current size of this global dictionary.	GB	A value close to the Maximum size measure is a cause of concern. Administrators should either consider increasing the size of the global dictionary or clean up the global dictionary to remove unwanted or outdated entries.
Percentage current size:	Indicates the current size of this global dictionary, in percentage.	Percent	
Maximum size:	Indicates the maximum size allocated to this global dictionary.	GB	

2.3.2 External Location Indices Consistency Test

The index server indexes and classifies documents and answers search queries. These indexes are stored in the relevant directories. If the location of the directories are altered, then the SAP TREX engine server may not be able to retrieve the indexes available in those directories. This in turn, may affect the search capabilities of the SAP TREX search engine server. Therefore, it is necessary to keep a constant vigil on the number of indexes that are present in the location that is other than the default directory. The **External Location Indices Inconsistency** test helps administrators in this regard.

This test monitors the number of indexes that were available in a location other than the location of the default directory. In addition, this test reports the total number of indexes in the directory and the indexes that are valid.

Target of the test : A SAP TREX

Agent deploying the test : An internal agent

Outputs of the test : One set of results for the SAP TREX server that is to be 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	Specify the port at which the specified host listens. By default, this is 30001.
TREX Directory	Specify the full path to the SAP TREX instance that is to be monitored for e.g., the location can be specified as D:\usr\sap\TRX\TRX01.
Timeout	Specify the duration (in seconds) beyond which this test should time out in the Timeout text box. The default is 120 seconds.
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.

Measures made by the test

Measurement	Description	Measurement Unit	Interpretation
Inconsistent indices	Indicates the number of indexes that were inconsistent i.e., stored in a different base path location as against the location where the data resides.	Number	If the indexes are corrupted/inconsistent, then the indexes should be recreated in order to be accessed. Therefore, if too many indexes are corrupted/inconsistent, then the performance of the TREX server would be affected. The detailed diagnosis of this measure lists the name of the indexes that were inconsistent, the namespace corresponding to the indexes etc.
Valid indices	Indicates the number of indexes with the correct base path and namespace location.	Number	A high value is desired for this measure.
Total indices	Indicates the total number of indexes stored on the disk.	Number	

2.3.3 Fast Search Infrastructure Indices Check Test

Fast search infrastructure searches for SAP business objects and joins between tables in business objects. The Fast search infrastructure module in SAP environments is mainly used to store and manage data. When a SAP TREX engine searches for data available in the Fast search infrastructure module, the index corresponding to the data is retrieved first. If the index is corrupted or not available, then the data corresponding to that index may not be retrieved as expected and the search may eventually fail. To avoid such search engine failures, it is necessary to maintain the accuracy of the indexes corresponding to the Fast search infrastructure module. This is where the **Fast Search Infrastructure Indices Check** test helps!

This test monitors the indexes of the Fast search infrastructure module and reports the total number of indexes corresponding to the Fast search infrastructure module, the indexes that are valid and the indexes that are corrupt.

Target of the test : A SAP TREX

Agent deploying the test : An internal agent

Outputs of the test : One set of results for the SAP TREX server that is to be monitored

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 -** Specify the port at which the specified **HOST** listens. By default, this is 30001.
4. **TREX DIRECTORY -** Specify the full path to the SAP TREX instance that is to be monitored for e.g., the location can be specified as *D:\usr\sap\TRX\TRX01*.
5. **TIMEOUT -** Specify the duration (in seconds) beyond which this test should time out in the **TIMEOUT** text box. The default is 120 seconds.
6. **AUTOCORRECT INCONSISTENCIES -** By default, this flag is set to **No** indicating that the corrupt indexes of the Fast Search Infrastructure Module are not deleted, by default. Set this flag to **Yes** to automatically delete the corrupt indexes.
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.

Measures made by the test:

Measurement	Description	Measurement Unit	Interpretation
Corrupt indices:	Indicates the number of indexes that were corrupted.	Number	If the indexes are corrupted, then the indexes should be recreated in order to be accessed. Therefore, if too many indexes are corrupted, then the performance of the TREX server will be affected. The detailed diagnosis of this measure lists the indexes that were corrupted.
Valid indices:	Indicates the number of indexes in the Fast Search Infrastructure module that passed the consistency check.	Number	A high value is desired for this measure.
Total indices:	Indicates the total number of indexes created for the data stored in the Fast Search Infrastructure module.	Number	

2.3.4 Trex Indices Test

The index server indexes and classifies documents and answers search queries. The processing takes place in the engines that belong to the index server. There are the following engines:

- Search engine: This engine is responsible for standard search functions such as the exact, error-tolerant, linguistic, Boolean, and phrase searches.
- Text-mining engine: This engine is responsible for classification, searching for similar documents (‘See Also’ search), the extraction of key words, and so on.
- Attribute engine: This engine is responsible for searching for document attributes such as author, creation date, and change date.

This test auto-discovers the indices available in the SAP TREX server and reports the current status of each index. In addition, this test helps administrators figure out the effectiveness of each index.

Target of the test : A SAP TREX

Agent deploying the test : An internal agent

Outputs of the test : One set of results for each Index type:Index of the target SAP TREX server that is to be monitored

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 -** Specify the port at which the specified **HOST** listens.
4. **TREX DIRECTORY -** Specify the full path to the SAP TREX instance that is to be monitored for e.g., the location can be specified as *D:\usr\sap\TRX\TRX01*.
5. **TIMEOUT -** Specify the duration (in seconds) beyond which this test should time out in the **TIMEOUT** text box. The default is 120 seconds.
6. **INCLUDE INDICES -** Specify a comma-separated list of indices that should be included in the scope of monitoring. By default, *none* is specified in this text box. Alternately, if you wish to monitor indices of a similar pattern, then you can specify such patterns in this list.
7. **EXCLUDE INDICES -** Specify a comma-separated list of indices that should be excluded in the scope of monitoring. By default, *none* is specified in this text box. Alternately, if you wish to monitor indices of a similar pattern, then you can specify such patterns in this list.
8. **SEARCHABLE DOCS CUTOFF -** By default, when an index is queried, the number of searchable documents associated with the index can be figured out by the administrator. Mostly, administrators may be interested to monitor only the indices that are associated with a large number of searchable documents. Also, in environments where huge volume of indices are to be monitored, the performance overhead of the eG agent may increase considerably. Therefore, to cater to the needs of such administrators and to reduce the performance overhead of the eG agent, it is necessary to provide a cutoff value to the number of searchable documents associated with the indices that are to be monitored. By providing this cutoff value, the indices with the number of searchable documents with less than the cutoff limit would be excluded from the scope of monitoring of this test. By default, this cutoff limit is set to 100. If you have specified the indices to be monitored in the **INCLUDE INDICES** parameter, then this parameter would be void.

9. **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.

Measures made by the test:

<table>
<tr><th>Measurement</th><th>Description</th><th>Measurement Unit</th><th>Interpretation</th></tr>
<tr><td>Status:</td><td>Indicates the current status of this index.</td><td></td><td>The values reported by this measure and its numeric equivalents are mentioned in the table below:
<table>
<tr><th>Measure value</th><th>Numeric Value</th></tr>
<tr><td>Idle</td><td>0</td></tr>
<tr><td>Indexing</td><td>1</td></tr>
<tr><td>Preparing</td><td>2</td></tr>
<tr><td>Committing</td><td>3</td></tr>
<tr><td>Rolling back</td><td>4</td></tr>
<tr><td>Merging</td><td>5</td></tr>
<tr><td>Corrupt</td><td>6</td></tr>
<tr><td>Setting sparse</td><td>7</td></tr>
</table>
<b>Note:</b>
By default, this measure reports the</td></tr>
</table>

Measurement	Description	Measurement Unit	Interpretation
			Measure Values listed in the table above to indicate the current status of this index. The graph of this measure however, represents the status of a server using the numeric equivalents only - 0 to 7.
Search engine memory:	Indicates the memory used by the search engine to process the search queries from this index.	KB	
Attribute engine memory:	Indicates the memory used by the attribute engine to search for document attributes from this index.	KB	
Text mining memory:	Indicates the memory used by the text mining engine to search for key words in this index.	KB	
Total memory:	Indicates the total memory used by this index.	KB	
Search engine disc size:	Indicates the amount of disk space used by the search engine to process search queries from this index.	KB	
Attribute engine disc size:	Indicates the amount of disk space used by the attribute engine to search for document attributes from this index.	KB	

Measurement	Description	Measurement Unit	Interpretation
Text mining disc size:	Indicates the amount of disk space used by the text mining engine to search for key words in this index.	KB	
Total disc size:	Indicates the amount of disk space used by this index.	KB	
Updated documents:	Indicates the number of documents that were updated for this index.	Number	
Deleted documents:	Indicates the number of documents that were deleted for this index.	Number	
Optimized documents:	Indicates the number of documents that were optimized for this index.	Number	A low value is desired for this measure. If too many documents are optimized i.e., if the index server performs frequent optimization, then the index server also rewrites the index frequently affecting the performance of the index.
Invalidated documents:	Indicates the number of documents that were invalidated for this index.	Number	
Total indexed documents:	Indicates the total number of documents indexed for this index.	Number	

2.3.5 Trex Queues Test

The queue server coordinates the processing steps that take place during indexing. It collects incoming document, triggers preprocessing by the preprocessor, and further processing by the index server.

The queue server enables documents to be indexed asynchronously. This has the advantage that you can control the time of indexing. For example, you can schedule indexing for times when the system load is lower because there are fewer search queries.

In addition, the queue server can trigger index replication and integration of the delta index in the main index.

A queue is a store in which the queue server collects documents to be indexed and administration information.

There is a queue for each index. The queue server creates a queue automatically as soon as you create a new index. The queue has the same ID as the index.

A queue has several parameters that control the behavior of the queue. In particular, the parameters specify when the queue server triggers indexing and optimization of documents. It is important for performance reasons that you have optimum settings for the queue parameters.

This test auto-discovers the queue groups in the target SAP TREX server and reports the number of queues that were idle, suspended, stopped etc in each queue group. Using this test, administrators can figure out the exact number of queues that were idle and were taking too long to process the requests.

Target of the test : A SAP TREX

Agent deploying the test : An internal agent

Outputs of the test : One set of results for each queue group of the target SAP TREX server that is to be monitored

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 -** Specify the port at which the specified **HOST** listens.
4. **TREX DIRECTORY -** Specify the full path to the SAP TREX instance that is to be monitored for e.g., the location can be specified as *D:\usr\sap\TRX\TRX01*.
5. **INCLUDE QUEUE GROUP PATTERNS -** Specify a comma-separated list of patterns (identified on the queues) that should be included in the scope of monitoring of this test. By default, *none* is specified here.

6. **EXCLUDE QUEUE GROUP PATTERNS -** Specify a comma-separated list of patterns (identified on the queues) that should be excluded from the scope of monitoring of this test. By default, *none* is specified here.

7. **TIMEOUT -** Specify the duration (in seconds) beyond which this test should time out in the **TIMEOUT** text box. The default is 120 seconds.

8. **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.

Measures made by the test:

Measurement	Description	Measurement Unit	Interpretation
Idle queues	Indicates the number of queues that were idle in this queue group.	Number	The queue server has a status for each queue. If the status of the queue server is idle, then , it means that the queue server is collecting incoming documents and triggering their preprocessing.
Indexing queues	Indicates the number of queues that were in the indexing state in this queue group.	Number	If the queues are in indexing state, then it means that the queue server is transmitting documents to the index server and triggering indexing.
Optimizing queues:	Indicates the number of queues that were in the optimizing state in this queue group.	Number	If the queues are in optimizing state, then, it means that the queue has triggered optimization and is waiting for a response from

Measurement	Description	Measurement Unit	Interpretation
			the index server.
Suspend- in-progress queues	Indicates the number of queues that were in *Suspend- in- Progress* state in this queue group.	Number	
Suspended queues	Indicates the number of queues that were in Suspended state in this queue group.	Number	
Stopped queues	Indicates the number of queues that were in Stopped state in this queue group.	Number	
Invalid queues	Indicates the number of queues that were invalid in this queue group.	Number	
Repairing queues	Indicates the number of queues that were in Repairing state in this queue group.	Number	
Merging queues	Indicates the number of queues that were in Merging state in this queue group.	Number	
Replicating queues	Indicating the number of queues that were in Replicating state in this queue group.	Number	
Rolling back queues	Indicates the number of queues that were in Rolling Back state in this queue group.	Number	

Measurement	Description	Measurement Unit	Interpretation
Total queues	Indicates the total number of queues in this queue group.	Number	This measure will not appear for the individual queues being monitored by this test.
Queue documents to be preprocessed	Indicates the number of queue documents that were in the To Be Preprocessed state in this queue group.	Number	
Preprocessing queue documents	Indicates the number of queue documents that were in the Preprocessing state in this queue group.	Number	
Queue documents with preprocessing failure	Indicates the number of queue documents that were in the Preprocessing Failed state in this queue group.	Number	
Queue documents to be transmitted	Indicates the number of queue documents that were preprocessed and waiting to be transmitted for indexing in this queue group.	Number	
Queue documents being transmitted	Indicates the number of queue documents that were being transmitted to an index server from this queue group.	Number	
Queue documents with transmission failure	Indicates the number of queue documents that could not be transmitted to the index server from	Number	

Measurement	Description	Measurement Unit	Interpretation
	this queue group.		
Queue documents to be synchronized	Indicates the number of queue documents from this queue group that were waiting to be optimized.	Number	
Queue documents synchronizing	Indicates the number of queue documents from this queue group that were being optimized.	Number	
Queue documents undergoing synchronization failure:	Indicates the number of queue documents from this queue group that could not be optimized.	Number	
Queue documents synchronized	Indicates the number of queue documents from this queue group that were processed successfully.	Number	
Queue documents delayed	Indicates the number of queue documents from this queue group that were in Delayed state.	Number	
Invalid queue documents	Indicates the number of queue documents from this queue group that were invalid.	Number	

2.3.6 Trex Threads Test

How many requests a SAP TREX server can process depends upon the number of threads each service of the SAP TREX server has been configured with. Without adequate threads, the SAP TREX server may not be able to service all requests, thus resulting in delays in request processing, long pending request queues, and degraded TREX performance. To avoid such anomalies, administrators will have to track the usage of threads by the SAP TREX server, proactively detect a probable contention for threads, and accordingly decide whether/not to increase the size of the thread pool. To achieve this, administrators can take the help of the **Trex Threads** test.

This test indicates the total number of threads each service of the SAP TREX server can spawn, reports the number of threads that are running, idle, suspended etc for each service and thus reveals whether/not the SAP TREX server needs to be configured with more threads to handle its current task load and to shorten the request queue.

Target of the test : A SAP TREX

Agent deploying the test : An internal agent

Outputs of the test : One set of results for each *host:service* of the target SAP TREX server that is to be monitored

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 -** Specify the port at which the specified **HOST** listens.
4. **TREX DIRECTORY -** Specify the full path to the SAP TREX instance that is to be monitored for e.g., the location can be specified as *D:\usr\sap\TRX\TRX01*.
5. **TIMEOUT -** Specify the duration (in seconds) beyond which this test should time out in the **TIMEOUT** text box. The default is 120 seconds.
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.

Measures made by the test:

Measurement	Description	Measurement Unit	Interpretation
Stat request threads:	Indicates the number of threads that were processing the requests for this service.	Number	If the value of this measure is 0, then the incoming requests to the service will not be processed.
Idle threads:	Indicates the number of threads that were instantiated and were awaiting incoming requests for this service.	Number	
Inactive threads:	Indicates the number of threads that were inactive for this service.	Number	By comparing the value of this measure across services, you can identify the service on which too many threads are inactive.
Running threads:	Indicates the number of threads that were running and processing the requests for this service.	Number	The detailed diagnosis of this measure lists the caller, called party, user, method details, running duration etc.
Suspended threads:	Indicates the number of threads that were suspended for this service.	Number	
Sleeping threads:	Indicates the number of threads of this service that were sleeping.	Number	
Other threads:	Indicates the number of threads that were in other states corresponding to this service.	Number	Other states vary based on the service to which the thread belongs to. They could be any of the following states : AssignEvent, attributeengine,

Measurement	Description	Measurement Unit	Interpretation
			connected, ControlSearchCacheEvent, createIndex, executetreequery, fdir, fput, getMemWatchInfo, getSelectedQueues, incrsyncpoints, asxml/indexCellTable, loadhistory, asxml/multiIndexCellTab, optimize, optimize/load, ping, pingstat, _ _ PINGSTAT, pexec, preloadIndex, preloadIndex/load, prepareOptimize, reorg, search, asxml/searchCellTable, searching/lexicon, send, setIndexProperty, setIndexProperty/unload, set_index_ property, show, simplestat, splitIndex, storetrees, defineConcatAttribute
Total threads:	Indicates the total number of threads allocated for this service.	Number	

2.4 The Trex Services Layer

This layer helps administrators to figure out the following:

- The accessibility of the server when a HTTP request is simulated;
- The current configuration status of the RFC server;
- How many threads were configured for the RFC server?
- What is the current status of the RFC server?
- What is the current CPU and disk usage of the SAP TREX server?
- How well the search requests, indexing requests and merging requests were processed by the SAP TREX server?

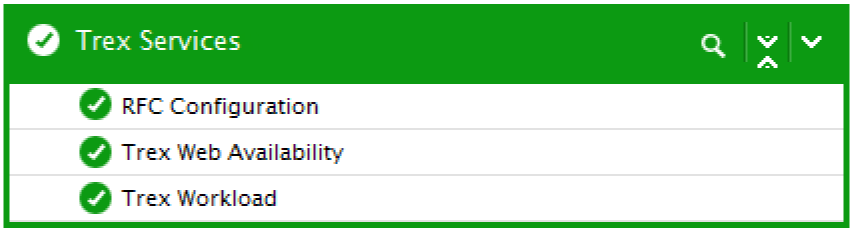


Figure 2.4: The tests mapped to the Trex Services layer

2.4.1 RFC Configuration Test

The RFC server is responsible for the communication between an SAP system and the TREX servers.

The SAP system sends requests to an RFC server using an SAP Gateway. The RFC server converts the requests to a TREX-internal format and then forwards them to the corresponding TREX servers. The TREX configuration depends on the structure of the SAP system. The TREX RFC Servers can be configured either in single-thread mode or in multi-thread mode. In the case of TREX RFC servers in single-thread mode, there are at least as many RFC servers (instances/processes) running on each TREX host as there are application servers. In the case of TREX RFC servers in multi-thread mode, exactly one RFC server (instance/process) runs on each TREX host and within this one RFC server, at least as many threads are started as there are applications servers. The TREX configuration depends on the structure of the SAP system. Each TREX host runs at least as many TREX RFC servers as there are application servers. If application servers are added to or removed from the SAP system, the TREX configuration should be changed accordingly. If the TREX configuration is not changed, then the communication between the RFC server and the SAP system may be disconnected. To avoid such disconnects, it is essential to monitor the configuration of the RFC server continuously. The **RFC Configuration** test helps administrators in this regard!

Using this test administrators can figure out the current configuration status of the RFC server with the target TREX server instance, the status of the RFC server and the number of threads configured on the RFC server.

Target of the test : A SAP TREX

Agent deploying the test : An internal agent

Outputs of the test : One set of results for the SAP TREX server that is to be monitored

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 -** Specify the port at which the specified **HOST** listens.
4. **TREX DIRECTORY -** Specify the full path to the SAP TREX instance that is to be monitored for e.g., the location can be specified as *D:\usr\sap\TRX\TRX01*.
5. **TIMEOUT -** Specify the duration (in seconds) beyond which this test should time out in the **TIMEOUT** text box. The default is 120 seconds.
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.

Measures made by the test:

Measurement	Description	Measurement Unit	Interpretation
RFC configuration status:	Indicates the current configuration status of the RFC server with the SAP TREX instance.		The values reported by this measure and its numeric equivalents are mentioned in the table below: <table><tr><th>Measure value</th><th>Numeric Value</th></tr><tr><td>Success</td><td>0</td></tr></table>

<table>
<tr><th>Measurement</th><th>Description</th><th>Measurement Unit</th><th>Interpretation</th></tr>
<tr><td></td><td></td><td></td><td>
<table>
<tr><th>Measure value</th><th>Numeric Value</th></tr>
<tr><td>Warning</td><td>1</td></tr>
<tr><td>Error</td><td>2</td></tr>
</table>
<b>Note:</b>
By default, this measure reports the Measure Values listed in the table above to indicate the current configuration status of the RFC server with the SAP TREX instance. The graph of this measure however, represents the status of a server using the numeric equivalents only - 0 to 2.
</td></tr>
<tr><td>RFC server status:</td><td>Indicates the current state of the RFC server i.e., indicates whether the RFC server and the associated threads of the server are configured to run properly.</td><td></td><td>
The values reported by this measure and its numeric equivalents are mentioned in the table below:
<table>
<tr><th>Measure value</th><th>Numeric Value</th></tr>
<tr><td>Success</td><td>0</td></tr>
<tr><td>Warning</td><td>1</td></tr>
<tr><td>Error</td><td>2</td></tr>
</table>
<b>Note:</b>
By default, this measure reports the Measure Values listed in the table above to indicate the current configuration status of the RFC server with the
</td></tr>
</table>

Measurement	Description	Measurement Unit	Interpretation
			SAP TREX instance. The graph of this measure however, represents the status of a server using the numeric equivalents only - 0 to 2. The detailed diagnosis of this measure lists the errors encountered by the RFC server.
Reachable destinations:	Indicates the number of RFC threads that are configured for the RFC server.	Number	

2.4.2 Trex Web Availability Test

This test emulates a user accessing the TREX web server via HTTP(S), and reports whether that server is accessible or not.

Target of the test : A SAP TREX

Agent deploying the test : An internal agent

Outputs of the test : One set of results for the target SAP TREX server being monitored

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 -** Specify the port at which the specified **HOST** listens.
4. **TREX DIRECTORY -** Specify the full path to the SAP TREX instance that is to be monitored for e.g., the location can be specified as *D:\usr\sap\TRX\TRX01*.
5. **TIMEOUT -** Specify the duration (in seconds) beyond which this test should time out in the **TIMEOUT** text box. The default is 120 seconds.
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.

Measures made by the test:

Measurement	Description	Measurement Unit	Interpretation
HTTP Code:	Indicates the response returned by the test for the simulated HTTP request.	Number	The detailed diagnosis of this measure lists the response code and its corresponding response.

2.4.3 Trex WorkLoad Test

Workload analysis for a SAP TREX server involves:

- Determining the number of search requests, indexing requests, index merging requests serviced per minute;
- Understanding the time spent to service the search requests, indexing requests, merging requests etc;
- Knowing how well the index server is processing the requests;
- Determining how well the memory of each component of the SAP TREX (RFC server, Index server, Queue server) is utilized.

This not only reveals the current workload of the SAP TREX server, but also highlights the processing ability of the SAP TREX server, pinpoints bottlenecks in processing, and leads administrators to where these bottlenecks lie. To perform such detailed workload analysis, administrators can use the **Trex Workload** test.

This test reports the current CPU and disk space usage of the server to indicate its current load. In addition, the test reveals the number of search requests, indexing requests and merging requests that the server processes every second, so that administrators can understand how well the server handles the load and can accurately identify where bottlenecks lie. By comparing the CPU usage of the server with its processing ability, administrators can intelligently figure out if the server requires additional CPU resources for improved performance.

Target of the test : A SAP TREX

Agent deploying the test : An internal agent

Outputs of the test : One set of results for the SAP TREX server that is to be monitored

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 -** Specify the port at which the specified **HOST** listens.
4. **WSDL PORT** - This test uses the SAPControl web service to pull metrics on application and service status. To enable the test to communicate with the web service, you need to configure the test with the port number of the web service. Therefore, specify the port number of the SAPControl web service against **WSDL PORT**. To determine the exact port number of the SAPControl web service, you can look up the etc/services file on the SAP TREX being monitored. If the port number is not declared in the etc/services file, you can specify the default port number of the web service against **WSDL PORT**. If the web service is not SSL-enabled, then the default port number of the web service will be: 5<NR>13. Similarly, if the web service is SSL-enabled, then the default port number of the web service will be: 5<NR>14. <NR> in the port number refers to the system number of the SAP TREX server being monitored. The system number is an indicator of the TCP/IP port at which the SAP TREX server listens. For example, for a server that listens at port 3200, the system number will be ‘00’. Similarly, if the SAP server port is 3201, the system number will have to be specified as ‘01’. Accordingly, the default port number of an SSL-enabled SAPControl web service will be 50014 , if the system number is 00, or 50114, if the system number is 01.
5. **SSL -** Set this flag to **Yes**, if the SAPControl web service is SSL-enabled. Set this flag to **No**, if the SAPControl web service is not SSL-enabled.

Measures made by the test:

Measurement	Description	Measurement Unit	Interpretation
Search rate:	Indicates the number	Requests/min	This measure is a good indicator

Measurement	Description	Measurement Unit	Interpretation
	of search requests per minute.		of the load on the TREX server. The load on the server is directly proportional to the search time.
Indexing rate:	Indicates the number of indexing requests per minute.	Requests/minute	A low value is desired for this measure as frequent indexing may affect the performance of the TREX server.
Merge rate:	Indicates the number of requests received to merge the delta indexes to the corresponding main indexes per minute.	Requests/minute	Merging or integration of delta indices with main indices needs to be scheduled with consideration. It involves rewriting of all main index files. Duration can be from a few minutes to several hours. During the merge, the index server cannot index new documents and hence indexing requests have to wait until such time. For such reasons merges are not scheduled during business hours and only if the delta index reaches a certain size. Due to the performance impact of merging, the merge requests rate needs to be monitored and configuration changes can be made to reduce merging as needed.
Unload rate:	Indicates the number of requests received to unload the index attributes from the memory per minute.	Requests/minute	Index attributes can be preloaded into memory for faster index performance. However, too many indexes preloaded into memory may cause free memory shortage resulting in unload requests to unload

Measurement	Description	Measurement Unit	Interpretation
			attributes of some indexes. An increasing unload rate is a clear indicator of memory issues in TREX. In such case, either the memory can be increased or index preloading can be reduced. Too many unloads is said to lead to server instability issues.
Search time:	Indicates the amount of time spent per minute to service the search requests.	Milliseconds/minute	When compared with search rate this measure tells us whether any increased search time is simply due to increased search requests or due to slowness in the search performance
Index time:	Indicates the amount of time spent per minute to index the documents/data in the index server.	Milliseconds/minute	When compared with index rate this measure tells us whether any increased index time is simply due to increased index requests or due to slowness in the indexing. During times of increased merge activity, indexing time will also suffer as the indexing may have to wait till the merging is done for the indices.
Merge time:	Indicates the amount of time spent to merge the delta indexes with the corresponding main indexes	Milliseconds/.minute	When compared with merge rate this measure tells us whether any increased merge time is simply due to increased merge requests or due to slowness in the merge. Typically merges are configured when the delta index size reaches 500 mb.

Measurement	Description	Measurement Unit	Interpretation
			Apart from delta index size, merge time is also proportional to the main index size.
Index server ping time:	Indicates the time taken by the index server to respond to requests.	Milliseconds	Unlike a standard TCP ping, this measurement corresponds to the time taken by the index server to respond to a watchdog service, typically, the nameserver. Correspondingly, this time depends upon the load on the index server.
Index server threads:	Indicates the number of threads instantiated on the index server.	Number	
Index server handles:	Indicates the number of handles generated and used by the index server.	Number	This includes the number of open files, sockets and other unknown handles.
Index server memory:	Indicates the amount of memory used by the index server to process the requests.	GB	If the index server is consuming high memory you can optimize the settings in TREXIndexServer.ini file. You can limit the max result set size for queries and parallel query execution. You can also consider turning off index statistics and index usage collection and using caches and compression.
Queue server memory:	Indicates the amount of memory used by the queue server.	GB	
RFC server memory:	Indicates the amount of memory used by the	GB	If this memory usage is high and you have many RFC servers

Measurement	Description	Measurement Unit	Interpretation
	RFC server.		running in single thread mode, consider switching to multi-thread mode to reduce the memory consumption
Overall memory:	Indicates the overall memory utilized by the TREX server.	GB	
Disk space used:	Indicates the disk space utilized by the TREX server.	GB	
CPU used:	Indicates the percentage of CPU utilized by the TREX server.	Percent	

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.

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

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