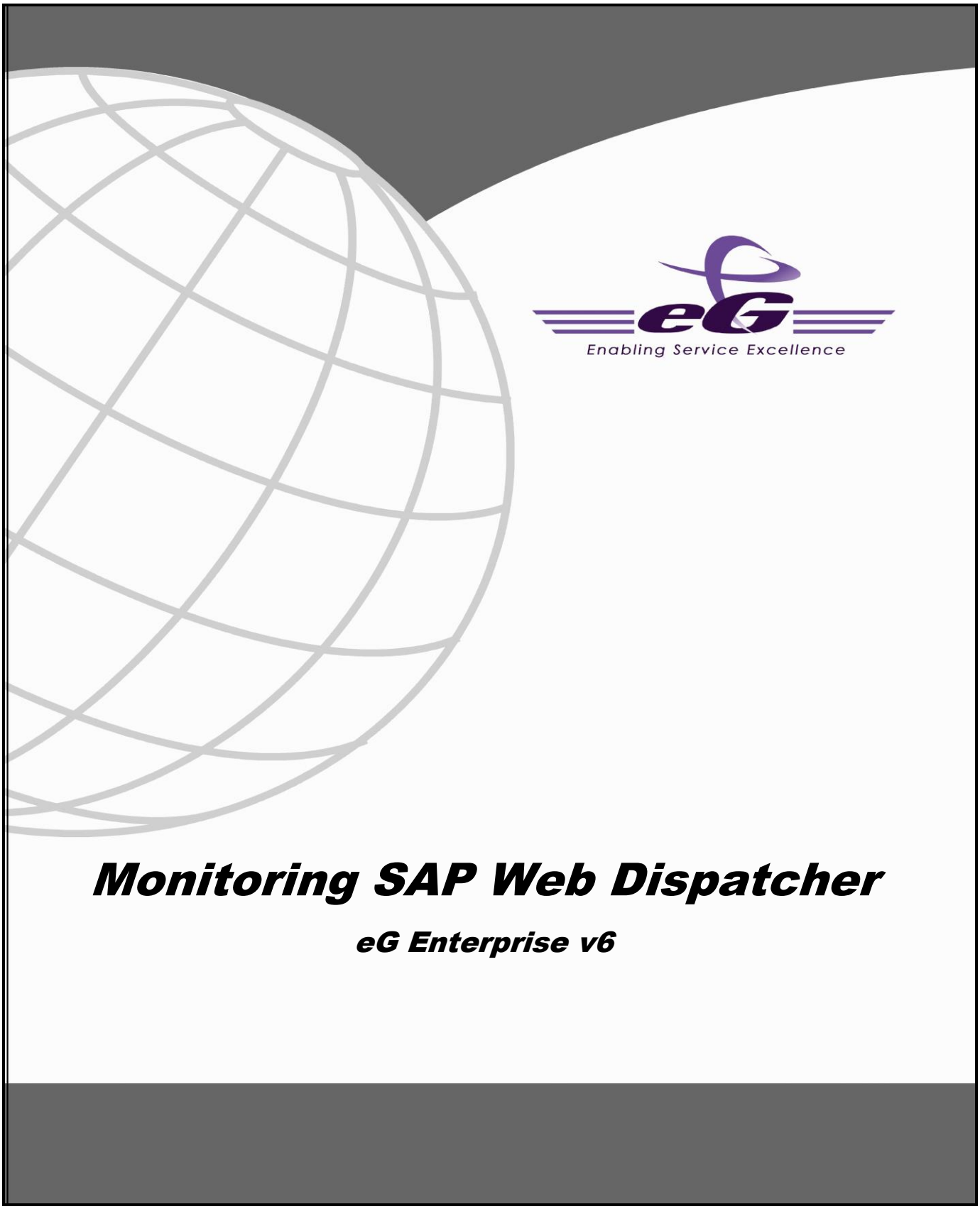


# Monitoring SAP Web Dispatcher

## eG Enterprise v6

**Restricted Rights Legend**

The information contained in this document is confidential and subject to change without notice. No part of this document may be reproduced or disclosed to others without the prior permission of eG Innovations, Inc. eG Innovations, Inc. makes no warranty of any kind with regard to the software and documentation, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose.

**Trademarks**

Microsoft Windows, Windows 2008, Windows 2012, Windows 7, Windows 8 and Windows 10 are either registered trademarks or trademarks of Microsoft Corporation in United States and/or other countries.

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

**Copyright**



The copyright in this document belongs to eG Innovations, Inc. Complying with all applicable copyright laws is the responsibility of the user.

# Table of Contents

# 1

# Introduction

The SAP Web Dispatcher lies between the Internet and your SAP system. It is the entry point for HTTP(s) requests into your system, which consists of one or more NetWeaver application servers. As a "software web switch", the SAP Web Dispatcher can reject or accept connections. When it accepts a connection, it balances the load to ensure an even distribution across the servers.

You can use the SAP Web Dispatcher in ABAP/Java systems and in pure Java systems, as well as in pure ABAP systems.

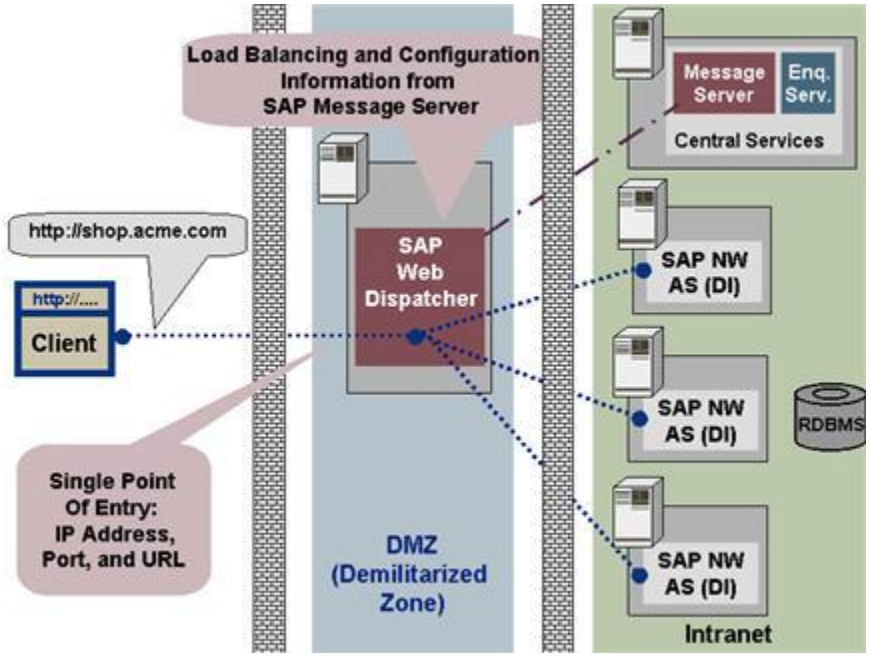


Figure 1: The SAP Web Dispatcher architecture

The SAP Web dispatcher performs the following tasks:

- *Selection of appropriate application server* (persistence with stateful applications, load balancing, ABAP or Java server).
- *Configuration for multiple systems* - You can place a SAP Web Dispatcher in front of multiple SAP systems, and configure which requests go to which system, or perform load balancing across system boundaries.
- *URL filtering* - You can define URLs that you want to be rejected, and by doing so restrict access to your system.
- *Web caching* – you can use the SAP Web Dispatcher as a Web Cache to improve the response times and to conserver the application server cache.
- *URL rewriting, manipulation of HTTP header fields* – The Web Dispatcher can manipulate inbound HTTP requests in general on the basis of defined rules.
- Depending on the *SSL* configuration you can forward, terminate, and (re)encrypt requests.

First of all, the SAP Web Dispatcher checks the type of the incoming request. The process flow described here works only if the request is not an administration request.

An HTTP request (or a decrypted HTTPS request) is assigned to a server in two stages.

First, the SAP Web Dispatcher decides whether the incoming HTTP request should be forwarded to an ABAP or a Java server. It ascertains a group of servers in the SAP system that could execute the request. It gets information about the groups from the back end (AS ABAP or AS Java), or from a file.

The load is then balanced within this group. When the load balancing process has decided on the server that the request is to be sent to, the SAP Web Dispatcher forwards it to the ICM of this application server.

If the request should be sent to an AS ABAP, then you must check whether a logon group (maintained in transaction SMLG) has been defined for this URL. The SAP Web dispatcher checks the list of logon groups to see if one has been defined. If it finds one, the load balancing must be executed within this logon group.

Both static and dynamic elements are used for load balancing with the SAP Web Dispatcher. The Web Dispatcher provides various procedures for load balancing.

Since with End-to-End SSL HTTPS requests the SAP Web Dispatcher is unable to read the URL, it can only distribute HTTPS requests in turn to the HTTPS-enabled servers in the system. This also takes into consideration the capacity of each server. To be able to process Java requests, each HTTPS-enabled server must have integrated the AS Java. The ICM of the server that receives the HTTPS request can decode the URL and then decide whether the request should be sent to AS ABAP or AS Java.

If the requests are not load balanced or if the SAP Web Dispatcher is not able to load balance the requests, then the requests may not be executed or may take too long to complete. To avoid such anomalies, eG Enterprise offers a specialized *SAP Web Dispatcher* monitoring model that monitors the current state of the SAP Web Dispatcher, the errors logged in trace log, security log etc, the current state and requests processed by each thread, the current state of each destination etc - and proactively alerts administrators to potential performance bottlenecks, so that administrators can resolve the issues well before end-users complain.

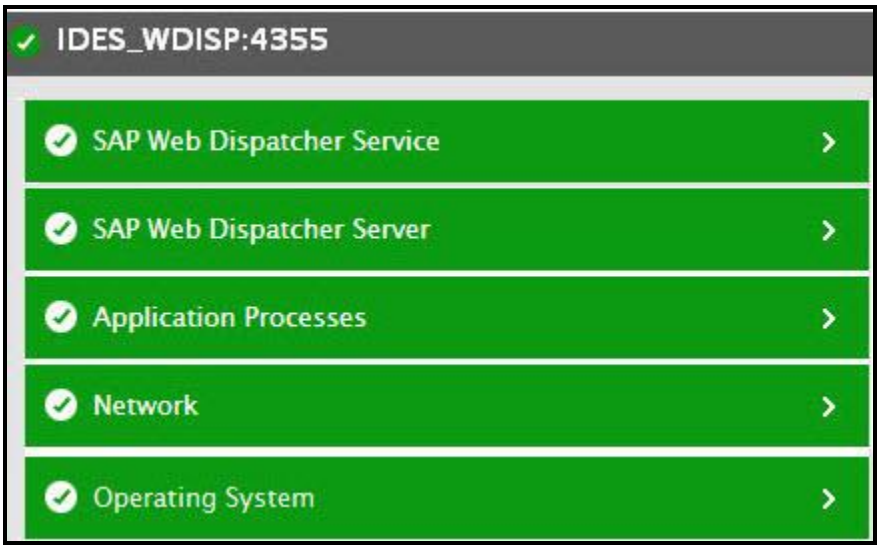


Figure 1.1: The layer model of SAP Web Dispatcher

Each layer of Figure 1.2 above is mapped to a variety of tests, each of which report a host of useful metrics pertaining to the SAP Web Dispatcher. Using these metrics, administrators can find quick and accurate answers to the following queries:

- What is the current state of the SAP Web Dispatcher?

- How many connections of each type are made to the SAP Web Dispatcher and how long did it take to establish a connection?
- How many requests were processed by each thread?
- What is the current state of each thread?
- What is the current operational state of the SAP Web Dispatcher?
- How many threads are created for processing the requests?
- How many connections are currently open?
- How many requests were awaiting the threads for processing?
- How many errors were encountered in the trace log, security log and http access log?
- Is the destination currently available?
- What is the current operational state of each destination?
- What is the capacity and load on each destination?
- How many stateful and stateless requests were processed on each destination?
- How long did it take for a request to reach the destination?
- How many HTTP and HTTPS connections were made to each destination?
- How many entries were stored in the SAP Web Dispatcher cache? Among them how many entries were invalid?
- What is the size of all the entries in the cache?

## 1.1 How does eG Enterprise Monitor the SAP Web Dispatcher?

By default, you can monitor and manage the SAP Web Dispatcher using the *Command line program*. To do so, you need to use the *wdispmon* executable. To monitor the SAP Web Dispatcher, you may need a user vested with *admin* privileges. By default, such a user is automatically created when the *SAP Web Dispatcher* is started using the *Bootstrap* option.

eG Enterprise employs an *agent-based* approach to monitor the SAP Web Dispatcher. In order to monitor the S*AP Web Dispatcher*, the user who is executing the *sapwebdisp* and *wdispmon* executables should possess 'admin' privileges.

Once the aforesaid requirements are fulfilled, the eG agent will report a plethora of useful metrics revealing the performance statistics of the SAP Web Dispatcher and present these performance statistics in the eG monitoring model using the hierarchical layer model representation of Figure 1.2.

The chapter that follows will discuss each layer of Figure 1.2 in great detail.

2

# Monitoring the SAP Web Dispatcher

This chapter deep dives into every layer of the *Docker* monitoring model, the tests mapped to each layer, and the measures every test reports.

## 2.1 The Application Processes Layer

The tests mapped to this layer reports the current state of the SAP Web Dispatcher.

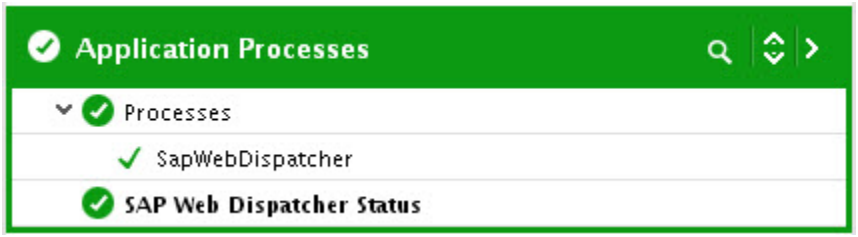


Figure 2: The tests mapped to the Application Processes layer

### 2.1.1 SAP Web Dispatcher Status Test

This test helps administrators to continuously monitor the SAP Web Dispatcher and reports the current state of the SAP Web Dispatcher.

| | |
|---|---|
| **Purpose** | Helps administrators to continuously monitor the SAP Web Dispatcher and reports the current state of the SAP Web Dispatcher |
| **Target of the test** | A SAP Web Dispatcher |
| **Agent deploying the test** | An internal agent |

<table>
<tr><td>Configurable parameters for the test</td><td colspan="3">1. TEST PERIOD – How often should the test be executed<br>2. HOST – Specify the host for which the test is to be configured<br>3. PORT - The port number at which the specified HOST listens. The default is 2375.<br>4. WEB DISPATCHER DIR – Specify the full path to the directory where the target SAP Web Dispatcher is installed. Ensure that the specified location contains the sapwebdisp and wdispmon executables.<br>5. PROFILE FILE – Specify the full path to the profile file with which the target SAP Web Dispatcher was started. The default name of the profile is sapwebdisp.pfl.<br>6. TIMEOUT - Specify the time for which the eG agent should wait for collecting the required metrics. By default, this is 60 seconds.<br>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.<br>The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled:<br>o The eG manager license should allow the detailed diagnosis capability<br>o Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.</td></tr>
<tr><td>Outputs of the test</td><td colspan="3">One set of results for the SAP Web Dispatcher to be monitored</td></tr>
<tr><td>Measurements made by the</td><td>Measurement</td><td>Measurement Unit</td><td>Interpretation</td></tr>
</table>

<table>
<tr>
<td>test</td>
<td>Status:<br>Indicates the current state of this SAP Web Dispatcher.</td>
<td></td>
<td>The values reported by this measure and their numeric equivalents are available in the table below:<br><table><tr><th>Measure Value</th><th>Numeric Value</th></tr><tr><td>Maintenance</td><td>0</td></tr><tr><td>Running</td><td>1</td></tr><tr><td>Soft Shutdown</td><td>2</td></tr><tr><td>Shutdown</td><td>3</td></tr></table>Note:<br>This measure reports the Measure Values listed in the table above while indicating the current state of the SAP Web Dispatcher. However, in the graph of this measure, this measure is indicated using only the Numeric Values listed above.<br>The detailed diagnosis of this measure if enabled, lists an error message if the state of the SAP Web Dispatcher is either Soft Shutdown or Shutdown.</td>
</tr>
</table>

## 2.2 The SAP Web Dispatcher Server Layer

The tests mapped to this layer reports the number of connections made to the dispatcher, the current state of each thread in the target SAP Web Dispatcher, the rate at which requests are made to the threads, the error messages recorded in the trace log etc..

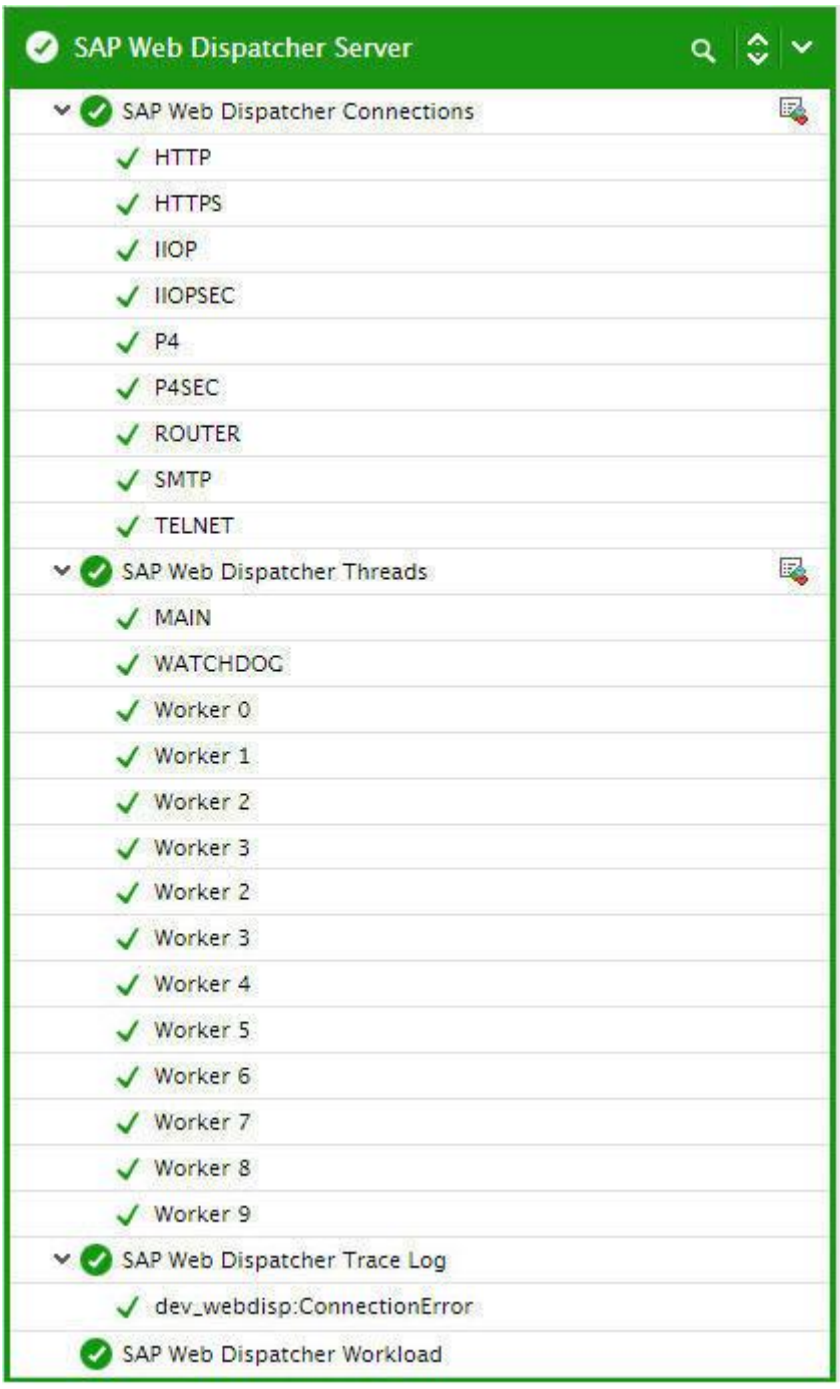


Figure 3: The tests mapped to the SAP Web Dispatcher Server layer

### 2.2.1 SAP Web Dispatcher Connections Test

As a "software web switch", the SAP Web dispatcher can reject or accept connections. When it accepts a connection, it balances the load to ensure an even distribution across the servers. The SAP Web Dispatcher therefore contributes to security and also balances the load in your SAP system. Often administrators are required to identify the number of connections made to the SAP Web Dispatcher and figure out how often such connections are made. The **SAP Web Dispatcher Connections** test helps administrators in this regard!

For each protocol type, this test reports the number of connections made to the target SAP Web Dispatcher and the maximum time taken by a connection. Using this test, administrators can figure out the protocol type through which maximum number of connections are made and identify the maximum time duration that is required for a connection to be established.

| | | | |
|---|---|---|---|
| **Purpose** | For each protocol type, this test reports the number of connections made to the target SAP Web Dispatcher and the maximum time taken by a connection. | | |
| **Target of the test** | A SAP Web Dispatcher | | |
| An internal/remote agent | An internal agent | | |
| **Configurable parameters for the test** | 1. **TEST PERIOD** – How often should the test be executed<br>2. **HOST** – Specify the host for which the test is to be configured<br>3. **PORT** - The port number at which the specified **HOST** listens. The default is 2375.<br>4. **WEB DISPATCHER DIR** – Specify the full path to the directory where the target SAP Web Dispatcher is installed. Ensure that the location specified contains the *sapwebdisp* and *wdispmon* executables.<br>5. **PROFILE FILE** – Specify the full path to the profile file with which the target SAP Web Dispatcher was started. The default name of the profile is *sapwebdisp.pfl*.<br>6. **TIMEOUT** - Specify the time for which the eG agent should wait for collecting the required metrics. By default, this is 60 seconds.<br>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.<br>The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled:<br>○ The eG manager license should allow the detailed diagnosis capability<br>○ Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0. | | |
| **Outputs of the test** | One set of results for each protocol type of connections made to the SAP Web Dispatcher that is being monitored | | |
| **Measurements made by the** | **Measurement** | **Measurement Unit** | **Interpretation** |

| | | | |
|---|---|---|---|
| **test** | **Number of connections:**<br>Indicates the total number of connections of this protocol type made to the SAP Web Dispatcher. | Number | This measure is a good indicator of the load on the SAP Web Dispatcher.<br>Comparing the value of this measure across the protocol types will enable you to figure out the protocol type that tops the maximum number of connections. |
| | **Maximum time for a connection:**<br>Indicates the maximum time taken to establish a connection of this type. | Mins | The Detailed diagnosis of this measure if enabled, lists the Top 3 slowest connections of each protocol type along with connection ID, Request type, address, connection time, timeouts etc. |

## 2.2.2 SAP Web Dispatcher Threads Test

The SAP Web dispatcher performs load balancing, and passes requests to Internet Communication Managers (ICMs) on the connected application servers rather than to work processes. Besides the pool of worker threads, which process incoming requests, the following ICM components are also implemented as threads:

- *Thread Control*: This thread accepts the incoming TCP/IP requests and creates (or wakes) a worker thread from the thread pool to process the request. From this point on, thread control initializes the connection info data.
- *Worker Threads*: These threads handle connection requests and responses. A worker thread contains an I/O handler for the network input and output, diverse plug-ins for the various supported protocols (HTTP, SMTP,…), which are required to be able to decide when the sent packet is finished (depends on the protocol).
- *Watchdog*: Usually, a worker thread waits for the response, regardless of whether the worker thread is a server or a client. If a timeout occurs, the watchdog takes on the task of waiting for the response. This makes the worker thread available for other requests. When the watchdog receives the response, it informs the thread control components, which then call a worker thread.
- *Signal Handler*: This thread processes signals sent from the operating system or from another process (for example, the dispatcher).
- *Connection Info*: This table contains information about the state of the connection, the memory pipes, and the plug-in data for every existing network connection.
- *Memory Pipes*: These memory-based communication objects are used to transfer data between the ICM and work processes (AS ABAP), or between the ICM and Java server processes (AS Java). There are four pipes for every connection: One data pipe per request and response and one out-of-band (OOP) pipe. The OOB pipe is used for control data.
- *Internet Server Cache*: The ICM contains another cache to enable a quick response to repeated requests. This cache is not shown in the graphic.

For each thread of the target SAP Web Dispatcher, this test reports the current state and the rate at which requests are processed by the threads.

| | |
|---|---|
| **Purpose** | For each thread of the target SAP Web Dispatcher, this test reports the current state and the rate at which requests are processed by the threads |
| **Target of the test** | A SAP Web Dispatcher |
| **Agent deploying the** | An internal agent |

<table>
<tr><td>test</td><td colspan="3"></td></tr>
<tr><td>Configurable parameters for the test</td><td colspan="3">1. TEST PERIOD – How often should the test be executed<br>2. HOST – Specify the host for which the test is to be configured<br>3. PORT - The port number at which the specified HOST listens. The default is 2375.<br>4. WEB DISPATCHER DIR – Specify the full path to the directory where the target SAP Web Dispatcher is installed. Ensure that the location specified contains the sapwebdisp and wdispmon executables.<br>5. PROFILE FILE – Specify the full path to the profile file with which the target SAP Web Dispatcher was started. The default name of the profile is sapwebdisp.pfl.<br>6. TIMEOUT - Specify the time for which the eG agent should wait for collecting the required metrics. By default, this is 60 seconds.<br>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.<br>The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled:<br>o The eG manager license should allow the detailed diagnosis capability<br>o Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.</td></tr>
<tr><td>Outputs of the test</td><td colspan="3">One set of results for each thread of the SAP Web Dispatcher being monitored</td></tr>
<tr><td rowspan="2">Measurements made by the test</td><td>Measurement</td><td>Measurement Unit</td><td>Interpretation</td></tr>
<tr><td>Requests:<br>Indicates the rate at which requests were processed by this thread during the last measurement period.</td><td>Requests/sec</td><td>This measure is a good indicator of the throughput of the web dispatcher threads.</td></tr>
</table>

| | | | |
|---|---|---|---|
| | **Status:**<br>Indicates the current state of this thread. | | The values reported by this measure and their numeric equivalents are available in the table below:<br>**Measure Value** / **Numeric Value**<br>Idle / 0<br>Running / 1<br>**Note:**<br>This measure reports the **Measure Value**s listed in the table above while indicating the current state of this thread. However, in the graph of this measure, this measure is indicated using only the **Numeric Value**s listed above.<br>The detailed diagnosis of this measure if enables lists the thread ID and the request type. |

## 2.2.3 SAP Web Dispatcher Workload Test

The ICM is implemented as an independent process and is started and monitored by the dispatcher. The ICM process uses a pool of worker threads to parallel process the load. This is why, if a sudden/consistent slowdown is noticed in a SAP system's interactions with the internet, the first place administrators need to check for inconsistencies is this thread pool. The absence of adequate threads in the pool can significantly impair the ICM's ability to uniformly distribute the request load across threads, thereby causing one/more threads be over-utilized; ultimately, this will result in a slowdown! Besides erratic thread pool usage, the sudden unavailability of the ICM and over-utilization of ICM connections can also cause disturbances in a SAP system's internet communications. To ensure that such anomalies are promptly captured and corrected, administrators should keep an eye on the accessibility of the ICM, its thread pool usage, and availability of ICM connections. This is where the **SAP Web Dispatcher Workload** test helps! This test periodically checks the availability, thread pool usage, and connection utilization of the ICM, and promptly reports the non-availability of the ICM, abnormal usage of worker threads by the ICM, and the over-utilization of ICM connections. This way, the test leads administrators to the probable causes for the breaks / slowness in the communication between the SAP system and the internet.

| | |
|---|---|
| **Purpose** | Reports the total number of images that are available on the Docker host/server. In addition, this test helps administrators to compare the numerical statistics of the images that are mapped to the containers and those that are not mapped to any container. Likewise, administrators can also be able to figure out the disk space utilization of images that are mapped to the containers and the disk space utilization of the images that are not mapped to the containers |
| **Target of the test** | A SAP Web Dispatcher |
| **Agent deploying the test** | An internal agent |

<table>
<tr><td>Configurable parameters for the test</td><td colspan="3">1. TEST PERIOD – How often should the test be executed<br>2. HOST – Specify the host for which the test is to be configured<br>3. PORT - The port number at which the specified HOST listens. The default is 2375.<br>4. WEB DISPATCHER DIR – Specify the full path to the directory where the target SAP Web Dispatcher is installed. Ensure that the location specified contains the sapwebdisp and wdispmon executables.<br>5. PROFILE FILE – Specify the full path to the profile file with which the target SAP Web Dispatcher was started. The default name of the profile is sapwebdisp.pfl.<br>6. TIMEOUT - Specify the time for which the eG agent should wait for collecting the required metrics. By default, this is 60 seconds.<br>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.<br>The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled:<br>o The eG manager license should allow the detailed diagnosis capability<br>o Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.</td></tr>
<tr><td>Outputs of the test</td><td colspan="3">One set of results for the target SAP Web Dispatcher that is being monitored</td></tr>
<tr><td rowspan="2">Measurements made by the test</td><td>Measurement</td><td>Measurement Unit</td><td>Interpretation</td></tr>
<tr><td>Status:<br>Indicates the current operational state of the SAP Web Dispatcher.</td><td></td><td>The values reported by this measure and their numeric equivalents are available in the table below:<br><table><tr><th>Measure Value</th><th>Numeric Value</th></tr><tr><td>Running</td><td>1</td></tr><tr><td>Maintenance</td><td>2</td></tr><tr><td>Soft Shutdown</td><td>3</td></tr></table>Note:<br>This measure reports the Measure Values listed in the table above while indicating the current operational state of the SAP Web Dispatcher. However, in the graph of this measure, this measure is indicated using only the Numeric Values listed above.</td></tr>
</table>

| | | | |
|---|---|---|---|
| | **Current threads:**<br>Indicates the total number of threads created from the thread pool of the SAP Web Dispatcher for processing requests. | Number | This measure is a good indicator of the load on the SAP Web Dispatcher. |
| | **Percentage current threads:**<br>Indicates the ratio of the total number of threads created to the maximum thread setting. | Percent | An unusually heavy network load causes the web dispatcher to continue creating threads until the maximum threads setting (icm/max_threads profile parameter) is reached, i.e., the percentage approaches 100. Once the limit is reached, the requests are queued and may be dropped due to queue full or timeouts. If this high thread usage is expected, then the maximum threads setting is low. Each operating system has an upper limit on the absolute maximum number of threads that its process can spawn depending upon its kernel parameters and available stack size. Please refer to kernel parameter documentation for your OS if you want to increase the maximum threads setting beyond the default limit set by your OS. |
| | **Used connections:**<br>Indicates the number of currently open connections. | Number | The number of simultaneously open connections and their sockets can be deduced from this measure. |
| | **Percentage connections used:**<br>Indicates the percentage of currently open connections i.e., the ratio of the number of connections used to the maximum connection setting. | Percentage | Each request needs a thread and a connection for processing. If this value is consistently high, it means that there are many requests awaiting further processing. Maximum connections setting (icm/max_conn profile parameter) is further restricted by maximum sockets setting(icm/max_sockets profile parameter) which in turn is further restricted by a corresponding OS setting. The web dispatcher needs 2 sockets per connection. |
| | **Requests in queue:**<br>Indicates the number of requests that are awaiting the threads for processing. | Number | |
| | **Percentage requests in queue:**<br>Indicates the ratio of the number of requests in the queue to the maximum queue setting. | Percentage | If this value is consistently high, it is possible that incoming requests are dropped due to insufficient space in the request queue. If this is expected, consider increasing the request queue size using the icm/req_queue_len profile parameter for the web dispatcher. |

### 2.2.4 SAP Web Dispatcher Trace Log Test

The SAP Web Dispatcher, automatically creates a **dev_webdisp_log** in the **SAP WEB DISPATCHER** directory to which errors/warnings captured during installation, configuration and control (starting/stopping) of the SAP Web Dispatcher are recorded. By periodically parsing this **dev_webdisp_log** and retrieving the details of error/warning events logged therein, you can understand which activities failed and why. This test helps you do just that!

| | |
|---|---|
| **Purpose** | By periodically parsing the **dev_webdisp_log** and retrieving the details of error/warning events logged therein, you can understand which activities failed and why. |
| **Target of the test** | A SAP Web Dispatcher |
| **Agent deploying the test** | An internal agent |

| | |
|---|---|
| **Configurable parameters for the test** | 1. **TEST PERIOD** - How often should the test be executed<br>2. **HOST** - The host for which the test is to be configured.<br>3. **PORT** – The port at which the server listens<br>4. **ALERTFILE -** Specify the full path to the log file to be monitored. Since the **dev_webdisp** is by default found in the **SAP WEB DISPATCHER** directory, the same is displayed here by default.<br>Also, instead of a specific log file path, the path to the directory containing the **dev_webdisp** files can be provided - eg., */opt/egurkha/manager/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 string *dev_webdisp*, the parameter specification can be, */opt/egurkha/manager/*dev_webdisp**. 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.<br>Every time this test is executed, the eG agent verifies the following:<br>➢ Whether any changes have occurred in the size and/or timestamp of the log files that were monitoring during the last measurement period;<br>➢ Whether any new log files (that match the **ALERTFILE** specification) have been newly added since the last measurement period;<br>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).<br>5. **SEARCHPATTERN** - Enter the specific patterns of alerts to be monitored. The pattern should be in the following format: *<PatternName>:<Pattern>*, where *<PatternName>* is the pattern name that will be displayed in the monitor interface and *<Pattern>* is an expression of the form - *expr* or expr or *expr or expr*, etc. A leading '*' signifies any number of leading characters, while a trailing '*' signifies any number of trailing characters. Multiple search patterns can be specified as a comma-separated list. If you want all the messages in a log file to be monitored, then your specification would be: *<PatternName>:*.<br>By default, this test monitors *error* and *warning* events logged in the **dev_webdisp** file. Therefore, the default **SEARCHPATTERN** is set as: *ConnectionError:*ERROR*,warn:*WARN**. This indicates that by default, the test will monitor only those lines in the **dev_webdisp** which embed the strings *ERROR* and *WARN*.<br>6. **LINES** - Specify two numbers in the format x:y. This means that when a line in the alert file matches a particular pattern, then x lines before the matched line and y lines after the matched line will be reported in the detail diagnosis output (in addition to the matched line). The default value here is 0:0. Multiple entries can be provided as a comma-separated list. |

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

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

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

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

7. **EXCLUDEPATTERN** - Provide a comma-separated list of patterns to be excluded from monitoring in the **EXCLUDEPATTERN** text box. For example **critical*, *exception**. By default, this parameter is set to 'none'.
8. **UNIQUEMATCH** - By default, the **UNIQUEMATCH** parameter is set to **FALSE**, indicating that, by default, the test checks every line in the log file for the existence of each of the configured **SEARCHPATTERN**. By setting this parameter to **TRUE**, you can instruct the test to ignore a line and move to the next as soon as a match for one of the configured patterns is found in that line. For example, assume that *Pattern1:*fatal*,Pattern2:*error** is the **SEARCHPATTERN** that has been configured. If **UNIQUEMATCH** is set to **FALSE**, then the test will read every line in the log file completely to check for the existence of messages embedding the strings 'fatal' and 'error'. If both the patterns are detected in the same line, then the number of matches will be incremented by 2. On the other hand, if **UNIQUEMATCH** is set to **TRUE**, then the test will read a line only until a match for one of the configured patterns is found and not both. This means that even if the strings 'fatal' and 'error' follow one another in the same line, the test will consider only the first match and not the next. The match count in this case will therefore be incremented by only 1.
9. **ROTATINGFILE** - This flag governs the display of descriptors for this test in the eG monitoring console. This flag is set to **false** by default. In this case, the descriptors of the test will be of the following format: *<Alert_file_name>:<SearchPattern>*.

   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>*. Since the **ALERTFILE** parameter is set to */manager/logs/error_log* by default in the case of this test, setting the **ROTATINGFILE** to **true** will display descriptors in the following format in the eG monitoring console: */manager/logs/dev_webdisp:<SearchPattern>*.

   On the other hand, if this flag is set to **true** and the **ALERTFILE** parameter is set to the directory containing log files (say, */manager/logs*), then, the descriptors of this test will be displayed in the format: *Configured_directory_path:<SearchPattern>* - i.e., */manager/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>*. */opt/egurkha/manager/*dev_webdisp**, and **ROTATINGFILE** is set to **true**, then, your descriptor will be: **error_log*:<SearchPattern>*. In this case, the descriptor format will not change even if the **ROTATINGFILE** flag status is changed.

| | |
|---|---|
| | 10. **CASESENSITIVE** - This flag is set to **No** by default. This indicates that the test functions in a 'case-insensitive' manner by default. This implies that, by default, the test ignores the case of your **ALERTFILE** and **SEARCHPATTERN** specifications. If this flag is set to **Yes** on the other hand, then the test will function in a 'case-sensitive' manner. In this case therefore, for the test to work, even the case of your **ALERTFILE** and **SEARCHPATTERN** specifications should match with the actual.<br>11. **OVERWRITTENFILE** - 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 *dev_webdisp*. When a roll over occurs, the content of the file *dev_webdisp* will be copied to a file named *dev_webdisp.1*, and all new errors/warnings will be logged in *dev_webdisp*. In such a scenario, since the **OVERWRITTENFILE** flag is set to **false** by default, the test by default scans only *error_log.1* for new log entries and ignores *dev_webdisp*. On the other hand, if the flag is set to **true**, then the test will scan both *dev_webdisp* and *dev_webdisp.1* for new entries.<br>If you want this test to support the 'roll over' capability described above, the following conditions need to be fulfilled:<br>➢ 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.<br>➢ The roll over file name should be of the format: "<alertfile>.1", and this file must be in the same directory as the alertfile.<br>12. **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.<br>13. **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.<br>The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled:<br>o The eG manager license should allow the detailed diagnosis capability<br>o Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0. |
| **Outputs of the test** | One set of results for each error type detected in the target SAP Web Dispatcher being monitored |
| **Measurements made by the** | **Measurement** / **Measurement Unit** / **Interpretation** |

| | | | |
|---|---|---|---|
| **test** | **Recent errors:**<br>Indicates the number of errors of this type that occurred during the last measurement period. | Number | |

## 2.3 The SAP Web Dispatcher Service Layer

Using this layer, administrators can monitor the availability of each destination connected to the target SAP Web Dispatcher, the operational state, load on each destination, the total size of the cache entries, the invalid entries in the cache, the errors detected in the log files etc.

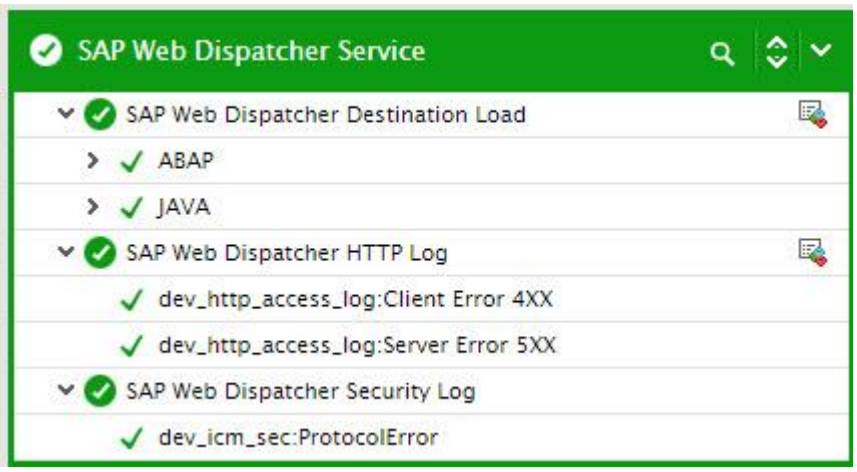


Figure 1.2: The tests mapped to the SAP Web Dispatcher Service layer

### 2.3.1 SAP Web Dispatcher Destination Status Test

The SAP Web Dispatcher can be set in front of multiple SAP and non-SAP Web Servers i.e., ABAP and J2EE destinations.

The Web Dispatcher then serves as the entry point for these systems/destinations and performs the following tasks:

- From the URL the Web Dispatcher identifies the system the request was destined for.
- The Web Dispatcher then performs load balancing for that particular system.

This test auto discovers the destinations that are connected to the SAP Web Dispatcher and reports the availability of each destination.

| | |
|---|---|
| **Purpose** | Auto discovers the destinations that are connected to the SAP Web Dispatcher and reports the availability of each destination |
| **Target of the test** | A SAP Web Dispatcher |
| **Agent deploying the test** | An internal agent |

<table>
<tr><td>Configurable parameters for the test</td><td colspan="3">1. TEST PERIOD – How often should the test be executed<br>2. HOST – Specify the host for which the test is to be configured<br>3. PORT - The port number at which the specified HOST listens. The default is 2375.<br>4. WEB DISPATCHER DIR – Specify the full path to the directory where the target SAP Web Dispatcher is installed. Ensure that the location specified contains the sapwebdisp and wdispmon executables.<br>5. PROFILE FILE – Specify the full path to the profile file with which the target SAP Web Dispatcher was started. The default name of the profile is sapwebdisp.pfl.<br>6. TIMEOUT - Specify the time for which the eG agent should wait for collecting the required metrics. By default, this is 60 seconds.<br>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.<br>The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled:<br>o The eG manager license should allow the detailed diagnosis capability<br>o Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.</td></tr>
<tr><td>Outputs of the test</td><td colspan="3">One set of results for each destination connected to the SAP Web Dispatcher being monitored</td></tr>
<tr><td rowspan="2">Measurements made by the test</td><td>Measurement</td><td>Measurement Unit</td><td>Interpretation</td></tr>
<tr><td>Is dispatcher available?:<br>Indicates whether/not this destination is available.</td><td></td><td>The values reported by this measure and their numeric equivalents are available in the table below:<br><table><tr><th>Measure Value</th><th>Numeric Value</th></tr><tr><td>No</td><td>0</td></tr><tr><td>Yes</td><td>1</td></tr></table>Note:<br>This measure reports the Measure Values listed in the table above while indicating whether/not this destination is available. However, in the graph of this measure, this measure is indicated using only the Numeric Values listed above.</td></tr>
</table>

## 2.3.2 SAP Web Dispatcher Destination Load Test

For each destination connected to the SAP Web Dispatcher, this test reports the current operational state, load, the number of connections, requests made to this destination etc.

| | | | |
|---|---|---|---|
| **Purpose** | For each destination connected to the SAP Web Dispatcher, this test reports the current operational state, load, the number of connections, requests made to this destination etc. | | |
| **Target of the test** | A SAP Web Dispatcher | | |
| **Agent deploying the test** | An internal agent | | |
| **Configurable parameters for the test** | 1. **TEST PERIOD** – How often should the test be executed<br>2. **HOST** – Specify the host for which the test is to be configured<br>3. **PORT** - The port number at which the specified **HOST** listens. The default is 2375.<br>4. **WEB DISPATCHER DIR** – Specify the full path to the directory where the target SAP Web Dispatcher is installed. Ensure that the location specified contains the *sapwebdisp* and *wdispmon* executables.<br>5. **PROFILE FILE** – Specify the full path to the profile file with which the target SAP Web Dispatcher was started. The default name of the profile is *sapwebdisp.pfl*.<br>6. **TIMEOUT** - Specify the time for which the eG agent should wait for collecting the required metrics. By default, this is 60 seconds.<br>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.<br>The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled:<br>o The eG manager license should allow the detailed diagnosis capability<br>o Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0. | | |
| **Outputs of the test** | One set of results for each *System:Destination* of the SAP Web Dispatcher being monitored | | |
| **Measurements made by the** | **Measurement** | **Measurement Unit** | **Interpretation** |

<table>
<tr>
<td>test</td>
<td>Status:<br>Indicates whether/not this destination is valid.</td>
<td></td>
<td>The values reported by this measure and their numeric equivalents are available in the table below:<br><table><tr><th>Measure Value</th><th>Numeric Value</th></tr><tr><td>No</td><td>0</td></tr><tr><td>Yes</td><td>1</td></tr></table>Note:<br>This measure reports the Measure Values listed in the table above while indicating whether /not this destination is valid. However, in the graph of this measure, this measure is indicated using only the Numeric Values listed in the above table.</td>
</tr>
<tr>
<td></td>
<td>Capacity:<br>Indicates the current capacity value for this destination.</td>
<td>Number</td>
<td>SAP Web dispatcher needs information about the capacity of a server in order to balance its workload in relation to the other servers. The capacity is used as an estimated value of the actual “power” of an application server.<br>Each application server has a capacity value for each of the services it provides (ABAP, Java). The number following <code>DIA=</code> or <code>LB=</code> is the capacity value for ABAP or Java.<br>For a destination in an ABAP system, the value of this measure is the number of configured work process and for a destination in a JAVA system, the value of this measure is the number of configured server nodes.<br>The value of this measure is taken into account by the SAP Web Dispatcher during load balancing.</td>
</tr>
<tr>
<td></td>
<td>Load:<br>Indicates the current load on this destination.</td>
<td>Number</td>
<td>The value of this measure and the Capacity measure is used during load balancing of the SAP Web Dispatcher.</td>
</tr>
<tr>
<td></td>
<td>Current HTTP connections:<br>Indicates the current number of HTTP connections to this destination.</td>
<td>Number</td>
<td>If the value of this measure is close to the value set against the wdisp/HTTP/max_pooled_con parameter, then it indicates that the SAP Web Dispatcher is handling too much of load and will not be able to accept new connections anymore.</td>
</tr>
</table>

| | | | |
|---|---|---|---|
| | **Percentage of HTTP connections:**<br>Indicates the percentage of HTTP connections to this destination. | Percentage | This measure is the ratio of the *Current HTTP connections* to the maximum number of HTTP connections setting. |
| | **Current HTTPS connections:**<br>Indicates the current number of HTTPS connections to this destination. | Number | If the value of this measure is close to the value set against the *wdisp/HTTPS/ max_pooled_con* parameter, then it indicates that the SAP Web Dispatcher is handling too much of load and will not be able to accept new connections anymore. |
| | **Percentage of HTTPS connections:**<br>Indicates the percentage of HTTPS connections to this destination. | Percentage | The value of this measure is the ratio of the *Current HTTPS connections* to the maximum number of HTTPS connection setting. |
| | **Stateless requests:**<br>Indicates the current number of stateless requests to this destination. | Number | Stateless requests are the requests that are assigned to the next application server in the queue in accordance with load balancing. |
| | **Request groups:**<br>Indicates the number of requests for groups in this destination. | Number | |
| | **Stateful requests:**<br>Indicates the current number of stateful requests to this destination. | Number | If a stateful request is made, the system selects the application server that should process the transaction. |
| | **Response time:**<br>Indicates the average time taken by a request to this destination. | Seconds | |
| | **Last ping time:**<br>Indicates the most recent response time for a ping request to this destination. | Seconds | The value of this measure can be used to detect overall load trends for the destination. A sudden/gradual increase in the value of this measure indicates bottlenecks in the destination. |

## 2.3.3 SAP Web Dispatcher Cache Test

The SAP Web dispatcher can be configured in such a way that the contents (static and dynamic) are stored in the cache, and the next request to this page can be satisfied quickly from the cache. The SAP Web dispatcher normally forwards incoming requests to an Internet Communication Manager (ICM), which is responsible for Web requests in SAP NetWeaver AS. The advantage of caching in the Web dispatcher over caching in the ICM is that the cache is located at the initial access point where all the requests arrive, and therefore all backend servers activated later and

their network connections are offloaded. Caching is deactivated in the SAP Web Dispatcher by default. You need to activate caching by specifying a URL prefix and cache directory in the parameter **icm/HTTP/server_cache_0** in the Web Dispatcher profile with which the SAP Web Dispatcher was started.

This test monitors the cache of the SAP Web Dispatcher and reports the total size of all cache entries, the total number of cache entries etc. Using this test, administrators can easily detect the number of invalid entries and remove them from the cache. This way, administrators can analyze how quickly the cache processes the requests. In the process, the test points to poor cache usage and processing bottlenecks, and reveals if the cache is poorly sized.

<table>
<tr><td>Purpose</td><td colspan="3">Monitors the cache of the SAP Web Dispatcher and reports the total size of all cache entries, the total number of cache entries etc. Using this test, administrators can easily detect the number of invalid entries and remove them from the cache</td></tr>
<tr><td>Target of the test</td><td colspan="3">A SAP Web Dispatcher</td></tr>
<tr><td>Agent deploying the test</td><td colspan="3">An internal agent</td></tr>
<tr><td>Configurable parameters for the test</td><td colspan="3">1. TEST PERIOD – How often should the test be executed<br>2. HOST – Specify the host for which the test is to be configured<br>3. PORT - The port number at which the specified HOST listens. The default is 2375.<br>4. WEB DISPATCHER DIR – Specify the full path to the directory where the target SAP Web Dispatcher is installed. Ensure that the location specified contains the sapwebdisp and wdispmon executables.<br>5. PROFILE FILE – Specify the full path to the profile file with which the target SAP Web Dispatcher was started. The default name of the profile is sapwebdisp.pfl.<br>6. TIMEOUT - Specify the time for which the eG agent should wait for collecting the required metrics. By default, this is 60 seconds.<br>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.<br>The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled:<br>o The eG manager license should allow the detailed diagnosis capability<br>o Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.</td></tr>
<tr><td>Outputs of the test</td><td colspan="3">One set of results for the SAP Web Dispatcher being monitored</td></tr>
<tr><td rowspan="2">Measurements made by the test</td><td>Measurement</td><td>Measurement Unit</td><td>Interpretation</td></tr>
<tr><td>Total entries:<br>Indicates the total number of entries in the cache.</td><td>Number</td><td>A high value is desired for this measure.</td></tr>
</table>

| | | | |
|---|---|---|---|
| | **Invalid entries:**<br>Indicates the number of expired or invalidated entries in the cache. | Number | If there are too many invalid entries, it indicates that the cache has become stale. If the value of this measure is gradually increasing, then it indicates that the cache is not sized adequately. The appropriate cache needs to be resized using the *icm/HTTP/server_cache_<x>/max_entries* and *icm/HTTP/server_cache_<xx>/size_MB* parameters |
| | **Percentage invalid entries:**<br>Indicates the percentage of expired or invalidated entries in the cache. | Percent | This measure is the ratio of the *Invalid entries* to *Total entries* measure.<br>A low value is desired for this measure. |
| | **Size:**<br>Indicates the total size of all cache entries in the cache. | KB | |

## 2.3.4 SAP Web Dispatcher HTTP Log Test

The SAP Web Dispatcher, automatically creates a **dev_http_access_log** in the **SAP WEB DISPATCHER** directory to which errors/warnings captured during installation, configuration and control (starting/stopping) of the SAP Web Dispatcher are recorded. By periodically parsing this **dev_http_access_log** and retrieving the details of error/warning events logged therein, you can understand which activities failed and why. This test helps you do just that.

| | |
|---|---|
| **Purpose** | By periodically parsing the **dev_http_access_log** and retrieving the details of error/warning events logged therein, you can understand which activities failed and why. |
| **Target of the test** | A SAP Web Dispatcher |
| **Agent deploying the test** | An internal agent |

| | |
|---|---|
| **Configurable parameters for the test** | 1. **TEST PERIOD** - How often should the test be executed<br>2. **HOST** - The host for which the test is to be configured.<br>3. **PORT** – The port at which the server listens<br>4. **ALERTFILE -** Specify the full path to the log file to be monitored. Since the **dev_http_access_log** is by default found in the **SAP WEB DISPATCHER** directory, the same is displayed here by default.<br>Also, instead of a specific log file path, the path to the directory containing the **error_log** files can be provided - eg., */opt/egurkha/manager/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 string *dev_http_access_log*, the parameter specification can be, */opt/egurkha/manager/*dev_http_access_log**. 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.<br>Every time this test is executed, the eG agent verifies the following:<br>➢ Whether any changes have occurred in the size and/or timestamp of the log files that were monitoring during the last measurement period;<br>➢ Whether any new log files (that match the **ALERTFILE** specification) have been newly added since the last measurement period;<br>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).<br>5. **SEARCHPATTERN** - Enter the specific patterns of alerts to be monitored. The pattern should be in the following format: *<PatternName>:<Pattern>*, where *<PatternName>* is the pattern name that will be displayed in the monitor interface and *<Pattern>* is an expression of the form - *expr* or expr or *expr or expr*, etc. A leading '*' signifies any number of leading characters, while a trailing '*' signifies any number of trailing characters. Multiple search patterns can be specified as a comma-separated list. If you want all the messages in a log file to be monitored, then your specification would be: *<PatternName>:**.<br>By default, this test monitors *error* and *warning* events logged in the **dev_http_access_log** file. Therefore, the default **SEARCHPATTERN** is set as: *error:*ERROR*,warn:*WARN**. This indicates that by default, the test will monitor only those lines in the **dev_http_access_log** which embed the strings *ERROR* and *WARN*.<br>6. **LINES** - Specify two numbers in the format x:y. This means that when a line in the alert file matches a particular pattern, then x lines before the matched line and y lines after the matched line will be reported in the detail diagnosis output (in addition to the matched line). The default value here is 0:0. Multiple entries can be provided as a comma-separated list. |

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

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

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

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

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

8. **UNIQUEMATCH** - By default, the **UNIQUEMATCH** parameter is set to **FALSE**, indicating that, by default, the test checks every line in the log file for the existence of each of the configured **SEARCHPATTERN**. By setting this parameter to **TRUE**, you can instruct the test to ignore a line and move to the next as soon as a match for one of the configured patterns is found in that line. For example, assume that *Pattern1:*fatal*,Pattern2:*error** is the **SEARCHPATTERN** that has been configured. If **UNIQUEMATCH** is set to **FALSE**, then the test will read every line in the log file completely to check for the existence of messages embedding the strings 'fatal' and 'error'. If both the patterns are detected in the same line, then the number of matches will be incremented by 2. On the other hand, if **UNIQUEMATCH** is set to **TRUE**, then the test will read a line only until a match for one of the configured patterns is found and not both. This means that even if the strings 'fatal' and 'error' follow one another in the same line, the test will consider only the first match and not the next. The match count in this case will therefore be incremented by only 1.

9. **ROTATINGFILE** - This flag governs the display of descriptors for this test in the eG monitoring console. This flag is set to **false** by default. In this case, the descriptors of the test will be of the following format: *<Alert_file_name>:<SearchPattern>*.

   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>*. Since the **ALERTFILE** parameter is set to */manager/logs/dev_http_access_log* by default in the case of this test, setting the **ROTATINGFILE** to **true** will display descriptors in the following format in the eG monitoring console: */manager/logs/ dev_http_access_log:<SearchPattern>*.

   On the other hand, if this flag is set to **true** and the **ALERTFILE** parameter is set to the directory containing log files (say, */manager/logs*), then, the descriptors of this test will be displayed in the format: *Configured_directory_path:<SearchPattern>* - i.e., */manager/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>*. */opt/egurkha/manager/*dev_http_access_log**, and **ROTATINGFILE** is set to **true**, then, your descriptor will be: **dev_http_access_log*:<SearchPattern>*. In this case, the descriptor format will not change even if the **ROTATINGFILE** flag status is changed.

<table>
<tr><td></td><td colspan="3">10. CASESENSITIVE - This flag is set to No by default. This indicates that the test functions in a 'case-insensitive' manner by default. This implies that, by default, the test ignores the case of your ALERTFILE and SEARCHPATTERN specifications. If this flag is set to Yes on the other hand, then the test will function in a 'case-sensitive' manner. In this case therefore, for the test to work, even the case of your ALERTFILE and SEARCHPATTERN specifications should match with the actual.<br><br>11. OVERWRITTENFILE - 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 dev_http_access_log. When a roll over occurs, the content of the file dev_http_access_log will be copied to a file named error_log.1, and all new errors/warnings will be logged in dev_http_access_log. In such a scenario, since the OVERWRITTENFILE flag is set to false by default, the test by default scans only dev_http_access_log.1 for new log entries and ignores dev_http_access_log. On the other hand, if the flag is set to true, then the test will scan both dev_http_access_log and dev_http_access_log.1 for new entries.<br><br>If you want this test to support the 'roll over' capability described above, the following conditions need to be fulfilled:<br>➢ 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.<br>➢ The roll over file name should be of the format: "<alertfile>.1", and this file must be in the same directory as the ALERTFILE.<br><br>12. 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.<br><br>13. 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.<br><br>The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled:<br>o The eG manager license should allow the detailed diagnosis capability<br>o Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.</td></tr>
<tr><td>Outputs of the test</td><td colspan="3">One set of results for each error type detected in the target SAP Web Dispatcher being monitored</td></tr>
<tr><td>Measurements made by the</td><td>Measurement</td><td>Measurement Unit</td><td>Interpretation</td></tr>
</table>

| | | | |
|---|---|---|---|
| **test** | **Recent errors:**<br>Indicates the number of errors of this type that occurred during the last measurement period. | Number | |

## 2.3.5 SAP Web Dispatcher Security Log Test

The SAP Web Dispatcher, automatically creates a **dev_icm_sec** in the **SAP WEB DISPATCHER** directory to which errors/warnings captured during installation, configuration and control (starting/stopping) of the SAP Web Dispatcher are recorded. By periodically parsing this **dev_icm_sec** and retrieving the details of error/warning events logged therein, you can understand which activities failed and why. This test helps you do just that.

| | |
|---|---|
| Purpose | By periodically parsing the dev_icm_sec and retrieving the details of error/warning events logged therein, you can understand which activities failed and why. |
| Target of the test | A SAP Web Dispatcher |
| Agent deploying the test | An internal agent |
| Configurable parameters for the test | 1. **TEST PERIOD** - How often should the test be executed<br>2. **HOST** - The host for which the test is to be configured.<br>3. **PORT** – The port at which the server listens<br>4. **ALERTFILE -** Specify the full path to the log file to be monitored. Since the **dev_icm_sec** is by default found in the **SAP WEB DISPATCHER** directory, the same is displayed here by default.<br>Also, instead of a specific log file path, the path to the directory containing the **error_log** files can be provided - e.g., */opt/egurkha/manager/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 string *dev_icm_sec*, the parameter specification can be, */opt/egurkha/manager/*dev_icm_sec**. 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.<br>Every time this test is executed, the eG agent verifies the following:<br>➢ Whether any changes have occurred in the size and/or timestamp of the log files that were monitoring during the last measurement period;<br>➢ Whether any new log files (that match the **ALERTFILE** specification) have been newly added since the last measurement period;<br>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).<br>5. **SEARCHPATTERN** - Enter the specific patterns of alerts to be monitored. The pattern |

should be in the following format: *<PatternName>:<Pattern>*, where *<PatternName>* is the pattern name that will be displayed in the monitor interface and *<Pattern>* is an expression of the form - *expr* or expr or *expr or expr*, etc. A leading '*' signifies any number of leading characters, while a trailing '*' signifies any number of trailing characters. Multiple search patterns can be specified as a comma-separated list. If you want all the messages in a log file to be monitored, then your specification would be: *<PatternName>:**.

By default, this test monitors *error* and *warning* events logged in the **dev_icm_sec** file. Therefore, the default **SEARCHPATTERN** is set as: *ProtocolError:*Error*,warn:*WARN**. This indicates that by default, the test will monitor only those lines in the **dev_icm_sec** which embed the strings *ERROR* and *WARN*.

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

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

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

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

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

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

8. **UNIQUEMATCH** - By default, the **UNIQUEMATCH** parameter is set to **FALSE**, indicating that, by default, the test checks every line in the log file for the existence of each of the configured **SEARCHPATTERN**. By setting this parameter to **TRUE**, you can instruct the test to ignore a line and move to the next as soon as a match for one of the configured patterns is found in that line. For example, assume that *Pattern1:*fatal*,Pattern2:*error** is the **SEARCHPATTERN** that has been configured. If **UNIQUEMATCH** is set to **FALSE**, then the test will read every line in the log file completely to check for the existence of messages embedding the strings 'fatal' and 'error'. If both the patterns are detected in the same line, then the number of matches will be incremented by 2. On the other hand, if **UNIQUEMATCH** is set to **TRUE**, then the test will read a line only until a match for one of the configured patterns is found and not both. This means that even if the strings 'fatal' and 'error' follow one another in the same line, the test will consider only the first match and not the next. The match count in this case will therefore be incremented by only 1.

9. **ROTATINGFILE** - This flag governs the display of descriptors for this test in the eG monitoring console. This flag is set to **false** by default. In this case, the descriptors of the test will be of the following format: *<Alert_file_name>:<SearchPattern>*.

   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>*. Since the **ALERTFILE** parameter is set to */manager/logs/dev_icm_sec* by default in the case of this test, setting the

**ROTATINGFILE** to **true** will display descriptors in the following format in the eG monitoring console: */manager/logs/error_log:<SearchPattern>*.

On the other hand, if this flag is set to **true** and the **ALERTFILE** parameter is set to the directory containing log files (say, */manager/logs*), then, the descriptors of this test will be displayed in the format: *Configured_directory_path:<SearchPattern>* - i.e., */manager/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>*. */opt/egurkha/manager/*dev_icm_sec**, and **ROTATINGFILE** is set to **true**, then, your descriptor will be: **dev_icm_sec*:<SearchPattern>*. In this case, the descriptor format will not change even if the **ROTATINGFILE** flag status is changed.

10. **CASESENSITIVE** - This flag is set to **No** by default. This indicates that the test functions in a 'case-insensitive' manner by default. This implies that, by default, the test ignores the case of your **ALERTFILE** and **SEARCHPATTERN** specifications. If this flag is set to **Yes** on the other hand, then the test will function in a 'case-sensitive' manner. In this case therefore, for the test to work, even the case of your **ALERTFILE** and **SEARCHPATTERN** specifications should match with the actual.
11. **OVERWRITTENFILE** - 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 *dev_icm_sec*. When a roll over occurs, the content of the file *dev_icm_sec* will be copied to a file named *error_log.1*, and all new errors/warnings will be logged in *dev_icm_sec*. In such a scenario, since the **OVERWRITTENFILE** flag is set to **false** by default, the test by default scans only *dev_icm_sec.1* for new log entries and ignores *dev_icm_sec*. On the other hand, if the flag is set to **true**, then the test will scan both *dev_icm_sec* and *dev_icm_sec.1* for new entries.

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

- The **ALERTFILE** parameter has to be configured only with the name and/or path of one/more alert files. File patterns or directory specifications should not be specified in the **ALERTFILE** text box.
- The roll over file name should be of the format: "<alertfile>.1", and this file must be in the same directory as the **ALERTFILE**.

12. **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.
13. **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

<table>
<tr><td></td><td colspan="3">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.<br><br>The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled:<br>o The eG manager license should allow the detailed diagnosis capability<br>o Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.</td></tr>
<tr><td>Outputs of the test</td><td colspan="3">One set of results for each error type detected in the target SAP Web Dispatcher being monitored</td></tr>
<tr><td rowspan="2">Measurements made by the test</td><td>Measurement</td><td>Measurement Unit</td><td>Interpretation</td></tr>
<tr><td>Recent errors:<br>Indicates the number of errors of this type that occurred during the last measurement period.</td><td>Number</td><td></td></tr>
</table>

# 3

# Conclusion

This document has clearly explained how eG Enterprise monitors the **SAP Web Dispatcher**. We can thus conclude that eG Enterprise, with its ability to provide in-depth insight into the performance of SAN storage infrastructures, is the ideal solution for monitoring such environments. For more information on eG Enterprise, please visit our web site at www.eginnovations.com or write to us at sales@eginnovations.com.