# Monitoring Siebel Web Server

eG Innovations Product Documentation

www.eginnovations.com

Total Performance Visibility

# Table of Contents

# Table of Figures

# Chapter 1: Introduction

Ever since business entities decided to go online with their service offerings, they have been having trouble dealing with a fast-expanding customer base and ever-mounting customer concerns. The lack of any efficient mechanism to manage the growing list of permanent/probable users to the service, caused the enterprise to lose millions; delays in follow-up calls lead to slow or no conversions, and poor customer support resulted in a loss of goodwill. That should explain why the service sector organizations providing business-critical services to end-users, have been turning to Customer Relationship Management (CRM) solutions like Siebel for help.

Siebel CRM-packaged business applications have become key enablers of an enterprise's customer-facing business processes. From tracking enquiries received from prospects to providing timely support to customers, the Seibel CRM modules automate the complete spectrum of activities that form part of an enterprise's marketing, sales, and support cycles. The wide capabilities of the Siebel solution demand a complex architecture; accordingly, you have a Siebel web client that front-ends requests to a Siebel web server, a Siebel gateway that grants the web requests access to the Siebel application servers, the Siebel application servers that process the requests by applying the business logic, and finally, the database server which stores and maintains the resultant data.

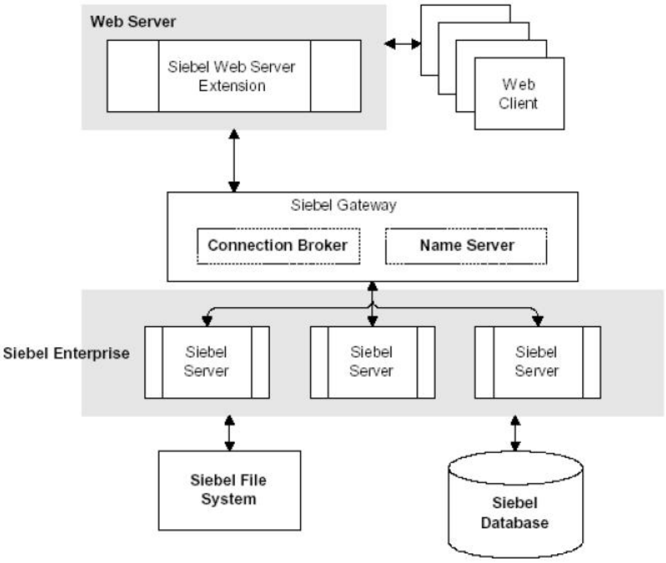


Figure 1.1: A high-level view of the Siebel application suite architecture

As multiple tiers of components are at work here, a problem with one tier/component can ripple and affect the performance of the dependent tiers. Siebel administrators therefore, often find it very difficult to determine where the real problem lies - is it with the Siebel web server? the Siebel gateway? the Siebel application server? or the database? The source of the problem has to be identified and necessary correction/optimization steps need to be taken to improve service performance and avoid service outages. What Siebel administrators need therefore, is an integrated solution that can monitor the entire chain of Siebel enterprise servers, taking into consideration the inter-dependencies that exist between them.

The eG Enterprise Suite, with its 100% web-based architecture and patented correlation and root-cause diagnosis capability, is ideal for monitoring Siebel environments. This solution offers exclusive monitoring models for analyzing the availability and overall health of every Siebel component. The data collectors employed by the suite extract a wide variety of performance statistics pertaining to the availability, responsiveness, session information, error logs and key tasks executing on these components. Besides measuring the health of the critical ingredients of a typical Siebel infrastructure, eG Enterprise also focuses on the performance of the operating systems that host the Siebel Enterprise components. Accordingly, a wealth of host-level performance information, which includes metrics on resource (CPU/memory/disk) usage by the host, key processes executing on the host, network availability and traffic to and from the host, etc., are collected. Using such extensive performance data, administrators can easily find answers to common Siebel Enterprise related queries like:

Moreover, the suite's end-to-end service monitoring capability and its patented root-cause diagnosis algorithm enable automatic correlation of the performance of the various Siebel components and quick and accurate problem isolation. By graphically representing a Siebel environment (see Figure 1.2), eG Enterprise enables administrators to quickly understand the interdependencies among Siebel components and their cause-effect relationships, and accurately judge root-cause of issues.

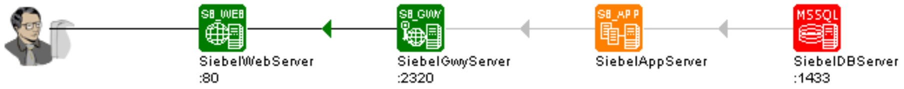


Figure 1.2: The topology model of the monitored Siebel environment

This document will engage you in an in-depth discussion of the monitoring model that eG Enterprise provides for the Siebel Web server, and the performance metrics that this model helps collect.

# Chapter 2: How does eG Enterprise Monitor Siebel Web Server?

eG Enterprise monitors the Siebel Web server in an agent based manner.

## 2.1 Configuring the Siebel Web Server

To enable the eG agent to collect the session-specific and other statistics from the web server, you need to configure the web server in the following manner:

- Edit the <SIEBEL_INSTALL_DIR>\sea<SIEBEL_VERSION>\SWEApp\BIN\eapps.cfg file on the Siebel web server host. For example, if Siebel 7.0.3 is installed in the **C** directory of a host, then the path to the configuration file will be as follows: C:\sea703\SWEApp\BIN\eapps.cfg.
- To enable the eG agent to extract session statistics from the web server, ensure that the **SessionMonitor** flag in the **eapps.cfg** file is set to **TRUE**.
- Similarly, by setting the **AllowStats** flag in the **eapps.cfg** file to **TRUE**, you can make sure that metric-collection is enabled on the Siebel web server to be monitored.
- Finally, save the file.

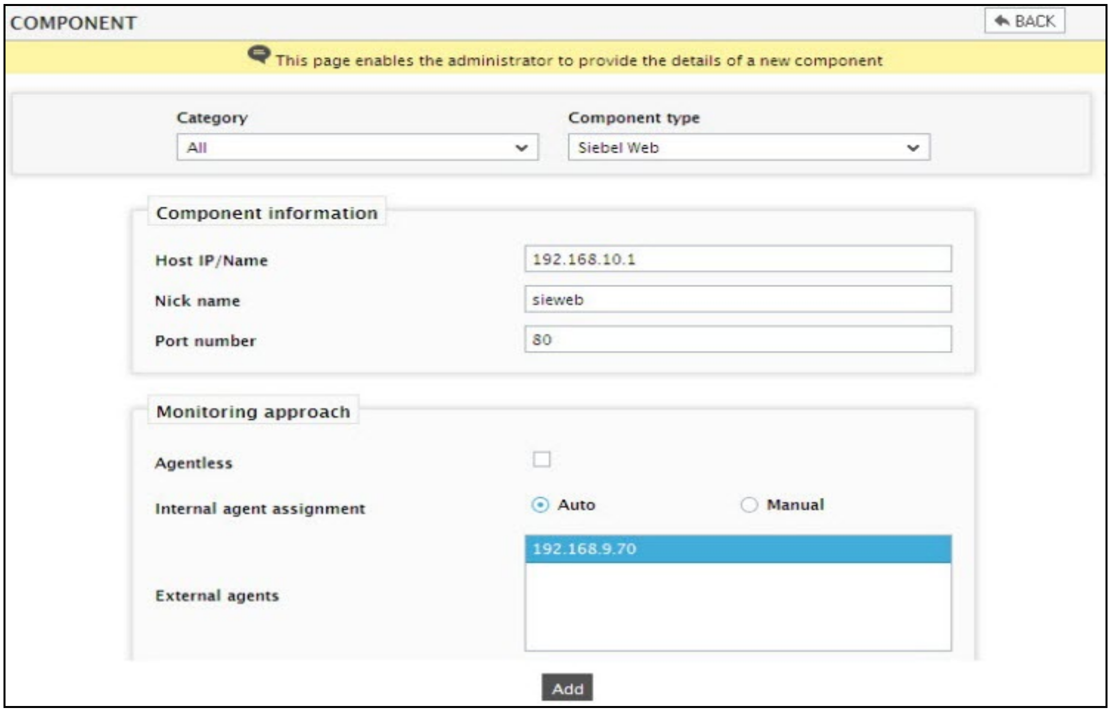
## 2.2 Managing the Siebel Web Server

eG Enterprise can automatically discover the Siebel Web server in the environment and also let you to add the Siebel Web server component if the server is not auto-discovered. The following steps explain you how to manage the server that is auto-discovered server and how to manually add the server using the eG administrative interface.

1. Log into the eG administrative interface.
2. If a Siebel Web server is already discovered, then directly proceed towards managing it using the **COMPONENTS – MANAGE/UNMANAGE** page (Infrastructure -> Components -> Manage/Unmanage).
3. However, if it is yet to be discovered, then run discovery (Infrastructure -> Components -> Discover) to get it discovered or add the component manually using the **COMPONENTS**page (Infrastructure -> Components -> Add/Modify). Remember that components manually added are managed automatically. Discovered components, however, are managed using the **COMPONENTS – MANAGE / UNMANAGE**page.

4. To add the component manually, follow the Components -> Add/Modify menu sequence in the **Infrastructure** tile of the **Admin** menu.

5. In the **COMPONENT** page that appears next, select *Siebel Web* as the **Component type**. Then, click the **Add New Component** button. This will invoke Figure 2.1.



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

Category: All

Component type: Siebel Web

Component information

Host IP/Name: 192.168.10.1

Nick name: sieweb

Port number: 80

Monitoring approach

Agentless: □

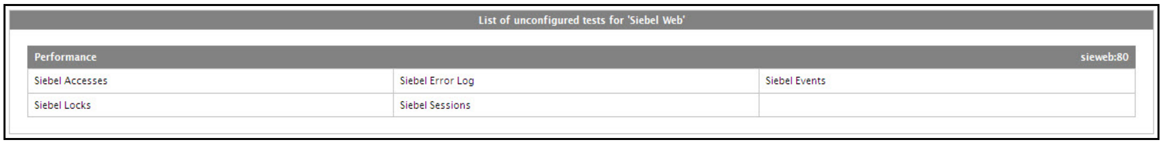
Internal agent assignment: ● Auto ○ Manual

External agents: 192.168.9.70

Figure 2.1: Adding a Siebel Web server

6. Specify the **Host IP/Name** and the **Nick name** of the Siebel Web server in Figure 2.1. Then, click the **Add** button to register the changes.

7. Next, try to sign out of the eG administrative interface. Figure 2.2 appears listing the tests to be configured for the managed Siebel Web server.



List of unconfigured tests for 'Siebel Web'

| Performance | | sieweb:80 |
|---|---|---|
| Siebel Accesses | Siebel Error Log | Siebel Events |
| Siebel Locks | Siebel Sessions | |

Figure 2.2: The list of unconfigured tests for the Siebel Web server

8. Click on any test in the list of unconfigured tests. Refer to **Monitoring the Siebel Application server** chapter to know more details on configuring these tests.

9. Finally, signout of the eG administrative interface.

# Chapter 3: Monitoring the Siebel Web Server

Using the Siebel web server extension running within, the Siebel Web Server maintains user sessions and manages the communication to Siebel Enterprise. The Siebel Web monitoring model (see Figure 3.1) that eG Enterprise prescribes for the Siebel Web server therefore, focuses on session behavior and related abnormalities.

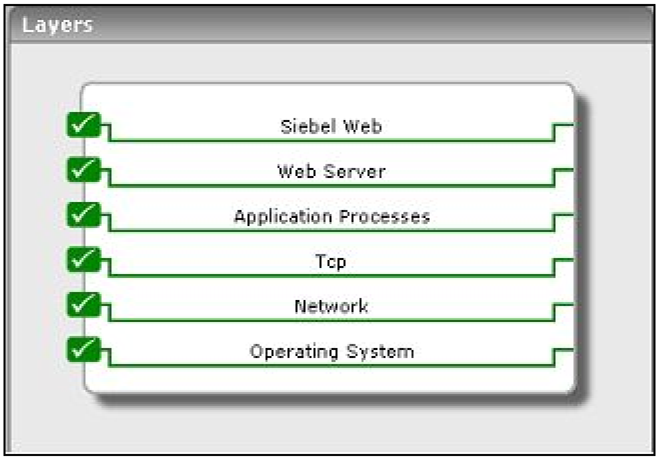


Figure 3.1: Layer model for a Siebel Web Server

To enable the eG agent to collect the session-specific and other statistics from the web server, you need to configure the web server using the steps explained in the Section **Chapter 2**.

Once monitoring is enabled on the web server, the eG agent can proceed to execute tests on the web server to determine the following:

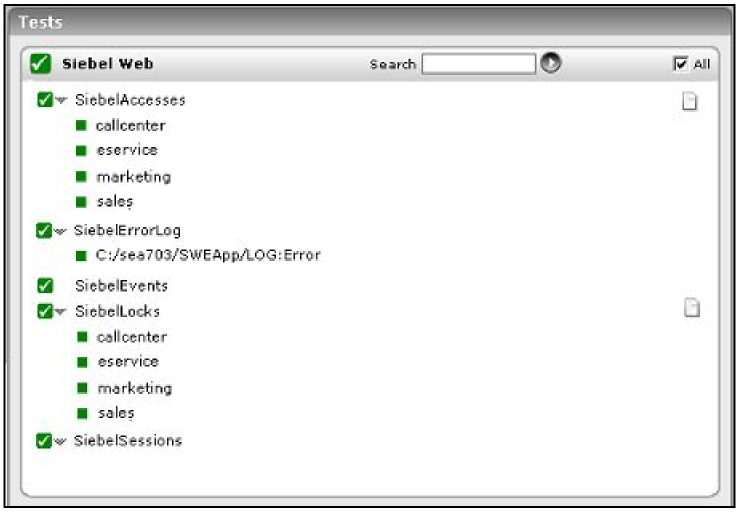
- Is the web server available?
- Is it responding quickly to client requests?
- Which are the most popular Siebel modules in terms of the number and duration of accesses?
- How many sessions are currently active on the web server?
- Did the sessions open too slowly? What about session closure? Did it also take too long?
- Were the sessions open for too long?
- Are there any anonymous sessions?
- Are too many errors logged in the error log?

Each of the layers depicted by Figure 3.1 above is mapped to one/more tests that an eG agent executes on the web server. The following sections deal with the **Siebel Web** layer only. For details on the **Web Server** layer, refer to the *Monitoring IIS Web Servers* document, and for details on the other layers, refer to the *Monitoring Unix and Windows Servers* document.

## 3.1 The Siebel Web Layer

Using the tests associated with it, the **Siebel Web** layer (see Figure 3.2) keeps a close watch on the accesses to the web server and the authenticated and anonymous sessions initiated on it, to reveal the following:

- The most popular application on the web server
- Errors (if any) that were recently encountered by the web server
- The session load on the web server
- The events triggered by session open/closure requests
- The number and duration of locks held by every monitored application on the web server



Tests

☑ Siebel Web  Search ☑ All

- ☑ SiebelAccesses
  - 🟩 callcenter
  - 🟩 eservice
  - 🟩 marketing
  - 🟩 sales
- ☑ SiebelErrorLog
  - 🟩 C:/sea703/SWEApp/LOG:Error
- ☑ SiebelEvents
- ☑ SiebelLocks
  - 🟩 callcenter
  - 🟩 eservice
  - 🟩 marketing
  - 🟩 sales
- ☑ SiebelSessions

Figure 3.2: Tests associated with the Siebel Web layer

### 3.1.1 Siebel Accesses Test

This test reports how often and for how long the configured application modules on the Siebel web server were accessed.

**Target of the test :** A Siebel web server

**Agent deploying the test :** An internal or remote agent

**Outputs of the test :** One set of results for each Siebel module monitored.

**Configurable parameters for the test**

| Parameter | Description |
|---|---|
| Test period | How often should the test be executed. |
| Host | The hostname (or IP address) of the Siebel web server. |
| Port | The port number on which the Siebel web server is listening. |
| URL | Specify the URL of the Siebel web server being monitored. |
| ApplicationName | Provide a comma-separated list of Siebel modules that need to be monitored. For example, *callcenter,eai,ecustomer*. |

**Measurements made by the test**

| Measurement | Description | Measurement Unit | Interpretation |
|---|---|---|---|
| Application hits | Indicates the number of current hits to this application. | Number | The value reported by this measure signifies the load to the specific Siebel application; it could also indicate how popular the application is. |
| Session life span | Indicates the duration of sessions to this application. | Secs | |

## 3.1.2 Siebel Error Log Test

All the events and errors that relate to the web server are tracked by the log file, along with the date, time and event for each log entry. Periodic monitoring of these log files can provide administrators with useful pointers to critical errors that might have affected the web server performance in recent times. This test reports the errors that were newly added to the web server log since the last measurement period.

**Target of the test :** A Siebel web server

**Agent deploying the test :** An internal agent

**Outputs of the test :** One set of results for every log file, log file directory, or *LogFilePath:PatternName* monitored on the Siebel web server .

### Configurable parameters for the test

| Parameter | Description |
|---|---|
| Test period | How often should the test be executed. |
| Host | The hostname (or IP address) of the Siebel web server. |
| Port | The port number on which the Siebel web server is listening. |
| AlertFile | Specify the path to the log file to be monitored. For eg., *C:/sea703/SWEBApp/LOG/Siebel_Web_log.txt*. Multiple log file paths can be provided as a comma-separated list.<br>Also, instead of a specific log file path, the path to the directory containing log files can be provided - eg., *C:/sea703/SWEBApp/LOG*. 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 'siebel' and 'log', the parameter specification can be, _C:/sea703/SWEBApp/LOG/*siebel*,C:/sea703/SWEBApp/LOG/*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>You can also configure the path in the following format:*Name@logfilepath*. Here, Name represents the display name of the path being configured. Accordingly, the parameter specification for the 'siebel' and 'log' example discussed above can be: _siebel@C:/sea703/SWEBApp/LOG/*siebel*,log@C:/sea703/SWEBApp/LOG/*log*_ . In this case, the display names 'siebel' and 'log' will alone be displayed as descriptors of this test.<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). |

| Parameter | Description |
|---|---|
| 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.<br>For example, say you specify Gen_errors:Generic* in the SearchPattern text box. This indicates that "Gen_errors" is the pattern name to be displayed in the monitor interface. "Generic*" indicates that the test will monitor only those lines in the log which start with the term "Generic".<br>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.<br>Multiple search patterns can be specified as a comma-separated list. For example: *Gen_errors:Generic*,Critical_errors:*Error**.<br>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*. |
| Lines | Specify two numbers in the format **x:y**. This means that when a line in the log 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.<br>If you give *1:1* as the value for LINES, then this value will be applied to all the patterns specified in the SearchPattern field. If you give *0:0*, *1:1* as the value for Lines and if the corresponding value in the SearchPattern filed is like Gen_errors:Generic*,Critical_errors:*Error*, then:<br>*0:0* will be applied to the Gen_errors:Generic*pattern<br>*1:1* will be applied to the Critical_errors:*Error*pattern |
| ExcludePattern | Provide a comma-separated list of patterns to be excluded from monitoring in the ExcludePattern text box. For example **critical*,*exception**. By default, this parameter is set to '*none*' |
| UniqueMatch | By default, 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 |

| Parameter | Description |
|---|---|
| | that line. For example, assume that *Pattern1:*Generic*,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 'Generic' 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 'Generic' 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. |
| RotatingFile | This flag governs the display of descriptors for this test in the eG monitoring console.<br>If this flag is set to true and the AlertFile text box contains the full path to a specific (log/text) file, then, the descriptors of this test will be displayed in the following format: *Directory_containing_monitored_file:<SearchPattern>*. For instance, if the AlertFile parameter is set to *c:\eGurkha\logs\syslog.txt*, and RotatingFile is set to **True**, then, your descriptor will be of the following format: *c:\eGurkha\logs:<SearchPattern>*. On the other hand, if the RotatingFile flag had been set to **False**, then the descriptors will be of the following format: *<FileName>:<SearchPattern>* - i.e., *syslog.txt:<SearchPattern>* in the case of the example above.<br>If this flag is set to **True** and the AlertFile parameter is set to the directory containing log files, then, the descriptors of this test will be displayed in the format: *Configured_ directory_path:<SearchPattern>*. For instance, if the AlertFile parameter is set to *c:\eGurkha\logs*, and rotatingfile is set to **True**, then, your descriptor will be: *c:\eGurkha\logs:<SearchPattern>*. On the other hand, if the RotatingFile parameter had been set to **False**, then the descriptors will be of the following format: *Configured_ directory:<SearchPattern>* - i.e., *logs:<SearchPattern>* in the case of the example above.<br>If this flag is set to **True** and the AlertFile parameter is set to a specific file pattern, then, the descriptors of this test will be of the following format: *<FilePattern>:<SearchPattern>*. For instance, if the AlertFile parameter is set to *c:\eGurkha\logs\*sys**, and RotatingFile is set to **True**, then, your descriptor will be: **sys*:<SearchPattern>*. In this case, the descriptor format will not change even if the RotatingFile flag status is changed. |
| 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. |

| Parameter | Description |
|---|---|
| | 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. |

**Measurements made by the test**

| Measurement | Description | Measurement Unit | Interpretation |
|---|---|---|---|
| Recent errors | Indicates the number of errors that were newly added to the log file when the test was last executed. | Number | The value of this measure is a clear indicator of the number of “new” errors that have occurred on the monitored Siebel web server. The detailed diagnosis of this measure provides the details of these new errors. |

### 3.1.3 Siebel Sessions Test

This test reports key statistics pertaining to the sessions on the Siebel web server.

**Target of the test :** A Siebel web server

**Agent deploying the test :** An internal or remote agent

**Outputs of the test :** One set of results for each Siebel web server monitored.

**Configurable parameters for the test**

| Parameter | Description |
|---|---|
| Test period | How often should the test be executed. |
| Host | The hostname (or IP address) of the Siebel web server. |
| Port | The port number on which the Siebel web server is listening. |
| URL | Specify the URL of the Siebel web server being monitored. |
| ApplicationName | Provide a comma-separated list of Siebel modules that need to be monitored. For example, *callcenter,eai,ecustomer*. |

| Parameter | Description |
|---|---|
| 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. |

### Measurements made by the test

| Measurement | Description | Measurement Unit | Interpretation |
|---|---|---|---|
| Current sessions | Indicates the number of sessions currently active on the Siebel web server. | Number | Since the user sessions include data on any user logged into the Siebel web server as well as the sessions created by the Siebel application, this measure is an accurate indicator of the direct loading on the web server from clients. As these user sessions run based on the Siebel server component task, the information on the user sessions can be viewed as either a user session or a task currently handled by the web server.<br>The detailed diagnosis of this measure, if enabled, lists all the current sessions to the Siebel web server. |
| New sessions | Indicates the number of sessions that newly opened on the Siebel Web server, the last time this test executed. | Number | Tracking the new sessions added over the monitoring interval enables administrators to be proactively alerted of any sudden, unusual increase in the load on the web server. By observing session load over a period of time, you |

| Measurement | Description | Measurement Unit | Interpretation |
|---|---|---|---|
| | | | can easily detect load trends, which in turn, will enable you to plan the capacity of the web server. |
| Avg session duration | Indicates the average duration of sessions. | Secs | An abnormal increase in the value of this measure could indicate that sessions are not getting closed properly. This hence necessitates further investigation. |

## 3.1.4 Siebel Events Test

This test monitors the session related events handled by the Siebel web server. The events are user specific actions, which help Siebel applications to respond in real time to user requests.

**Target of the test :** A Siebel web server

**Agent deploying the test :** An internal or remote agent

**Outputs of the test :** One set of results for every configured module on the web server monitored.

**Configurable parameters for the test**

| Parameter | Description |
|---|---|
| Test period | How often should the test be executed. |
| Host | The hostname (or IP address) of the Siebel web server. |
| Port | The port number on which the Siebel web server is listening. |
| URL | Specify the URL of the Siebel web server being monitored. |
| ApplicationName | Provide a comma-separated list of Siebel modules that need to be monitored. For example, *callcenter,eai,ecustomer*. |

**Measurements made by the test**

| Measurement | Description | Measurement Unit | Interpretation |
|---|---|---|---|
| Anonymous sessions requested | Indicates the number of anonymous session | Sessions | Ideally, the value reported for this |

| Measurement | Description | Measurement Unit | Interpretation |
|---|---|---|---|
| | requests handled by the Siebel web Server. | | measure should be low. A high value for this measure indicates that the number of anonymous users accessing the web server has increased. The benchmark set for optimizing the Siebel performance depends upon defining MaxTasks, MTServers and Anonuserpool values against the target no of users. Suppose if your target no of users is 4000 and you have defined MaxMTServer=MinMTServer to 58, the MaxTasks defined for this scenario could be 4600, taking into account the Anonymous user pool at any point in time to be 10% (400) of the Target users. An increase in the number of anonymous users could affect the ratio of threads per users, causing performance degradation in terms of longer response times. |
| Anonymous sessions removed | Indicates the number of anonymous sessions removed/terminated by the Siebel web server. | Sessions | A high value for this indicates that either sessions are being timed out or connectivity is not stable enough. |
| Anonymous sessions returned | Indicates the number of anonymous sessions returned to the web server. | Sessions | Anonymous sessions returned should be close to anonymous sessions requested. |
| Avg time for opening a session | This specifies the average amount of time spent by the server to open a session. | Secs | A steady/significant increase in the time taken to open sessions can point to probable issues, which, if left unresolved, can impair the end user experience. |
| Avg response time | Indicates the average | Secs | Ideally the value for this measure |

| Measurement | Description | Measurement Unit | Interpretation |
|---|---|---|---|
| | amount of time spent by the server to respond to the request. | | should be low. |
| Avg time for closing a session | Indicates the average amount of time taken by the sessions to close. | Secs | This event reflects the amount of time it takes to close a session. Closing the session might involve signaling to the session manager to close the session. The session manager might or might not close the TCP/IP connection.<br>If the value of this measure is very high, it indicates a bottleneck in session closure. The reasons for the same should hence be ascertained. |
| Avg request time | Indicates the average amount of taken to submit a request to the Siebel Server and to get a response back. | Secs | Ideally the value for this measure should be low. For example, if the user (on the browser) clicked on a button then the plug-in receives the request and invokes a service on the Siebel Server. The value for Request Time is the average amount of time for invoking that service. |

### 3.1.5 Siebel Locks Test

This test indicates the number and duration of locks on configured modules on the Siebel web server.

**Target of the test :** A Siebel web server

**Agent deploying the test :** An internal or remote agent

**Outputs of the test :** One set of results for each module configured for monitoring on the Siebel web server.

### Configurable parameters for the test

| Parameter | Description |
|---|---|
| Test period | How often should the test be executed. |
| Host | The hostname (or IP address) of the Siebel web server. |
| Port | The port number on which the Siebel web server is listening. |
| URL | Specify the URL of the Siebel web server being monitored. |
| ApplicationName | Provide a comma-separated list of Siebel modules that need to be monitored. For example, *callcenter,eai,ecustomer*. |

### Measurements made by the test

| Measurement | Description | Measurement Unit | Interpretation |
|---|---|---|---|
| Time taken to initialize locks | Represents the time that an identified user currently took to initialize locks on this module. | Secs | |
| Time taken to acquire anonymous locks | Represents the time that an anonymous user currently took to acquire locks on this module. | Secs | |
| Initialized locks | Indicates the number of locks that are currently initialized by identified users to this module. | Number | |
| Anonymous locks | Indicates the number of locks on this module that are currently held by anonymous users. | Number | |
| Avg time to initialize locks | Indicates the average time taken by identified users to initialize locks on this module. | Secs | |
| Avg time to acquire anonymous locks | Indicates the average time taken by anonymous users to acquire locks on this module. | Secs | |

## 3.2 Troubleshooting

If the tests related to the Siebel web server are not running, then first, try connecting to the following URL, and check whether it takes you to a page that lists the session-related and other web server-specific metrics for a configured ApplicationName: http://<IPoftheSibelWebServer>/<applicationnameconfiguredforthetest>/_stats.swe?verbose=high. For instance, if the IP address of the Siebel web server is, 192.168.10.12, and the ApplicationNameconfigured for a Siebel web server-related test is **callcenter**, the URL will be: http://192.168.10.12/callcenter/_stats.swe?verbose=high.

If the URL does not result in the display of the desired web page, then proceed to check whether the **AllowStats** and **SessionMonitor** flags in the **eapps.cfg** file (in the <SIEBEL_INSTALL_DIR>\sea<SIEBEL_VERSION>\SWEApp\BIN directory) are set to **TRUE**. To know how, refer to Section **Chapter 2**.

Also, you can verify the values reported by the tests associated with the Siebel application server component, using the **srvrmgr.exe** in the <SIEBEL_INSTALL_DIR>\sea<Siebel_version>\BIN directory. The syntax for the command is:

```
srvrmgr.exe /g <IPoftheGatewayServer> /e <SiebelEnterpriseServerNameconfiguredforthetest>
/u <UsernameoftheSiebelAdministrator> /p <PasswordoftheSiebelAdministrator> /c "<sub-
command>"
```

For instance, to check whether the statistics reported by the SiebelTasks test are accurate or not, do the following:

- Go to the command prompt of the Siebel application server.
- Switch to the <SIEBEL_INSTALL_DIR>\sea<Siebel_version>\BIN directory.
- Assume that the SiebelTasks test takes the following parameters:

  **GATEWAYSERVER -** 192.168.10.58

  **ENTERPRISESERVER -** siebel

  **USERNAME -** sadmin

  **PASSWORD -** sadmin

- Then, execute the following command on it:

  ```
  srvrmgr.exe /g 192.168.10.58 /e siebel /u sadmin /p sadmin /c "list tasks"
  ```

  where "list tasks" is the sub-command that is executed for viewing task-related metrics.

  Similarly, the command for the **Siebel Stats** test, will be:

```
srvrmgr.exe /g 192.168.10.56 /e siebel /u sadmin /p sadmin /c "list stats"
```

The following command will have to be executed for viewing the list of object managers configured on the Siebel server:

```
srvrmgr.exe /g 192.168.10.56 /e siebel /u sadmin /p sadmin /c "list comps"
```

- For the SiebelNet test, on the other hand, a utility named **visutl.exe** will have to be run from the <SIEBEL_INSTALL_DIR>\sea<SIEBEL_VERSION>\SWEApp\BIN directory. The syntax of this command is:

```
visutl.exe /u <SiebelAdministratorName> /p <SiebelAdministratorPassword> /c
<ODBCDatasourceNameconfiguredforthetest> /d <Tableownernameconfiguredforthetest> /n
<Nodenameconfiguredforthetest>
```

- For instance, assume that you want to verify the accuracy of the measures reported by the SiebelNet test, which takes the following parameters:

**USERNAME** - sadmin

**PASSWORD** - sadmin

**SIEBELDATASOURCENAME** - SiebSrvr_siebel

**TABLEOWNERNAME** - siebel

**NODENAME** - siebel

To achieve this, execute the following command from the <SIEBEL_ INSTALL_ DIR>\sea<SIEBEL_VERSION>\SWEApp\BIN directory:

```
visutl.exe /u sadmin /p sadmin /c SiebSrvr_siebel /d siebel /n siebel
```

# 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.