Siebel

System Monitoring and Diagnostics Guide

January 2024

Siebel
System Monitoring and Diagnostics Guide

January 2024

F84308-03

Author: Sukanya Mishra

Contents

Get Help

Preface

This preface introduces information sources that can help you use the application and this guide.

Using Oracle Applications

To find guides for Oracle Applications, go to the Oracle Help Center at *https://docs.oracle.com/*.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the *Oracle Accessibility Program website*.

Contacting Oracle

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit *My Oracle Support* or visit *Accessible Oracle Support* if you are hearing impaired.

Comments and Suggestions

Please give us feedback about Oracle Applications Help and guides! You can send an email to: *oracle_fusion_applications_help_ww_grp@oracle.com*.

1 What's New in This Release

What's New in Siebel System Monitoring and Diagnostics Guide, Siebel CRM 24.1 Update

The following information lists the changes in this revision of the documentation to support this release of the software.

What's New in Siebel System Monitoring and Diagnostics Guide, Siebel CRM 24.1 Update

Topic	Description
Example of Siebel Server Startup Log File	New ObjMgrGenericInfo event subtype. A new log category is introduced, that takes informational message from the ObjectMgrError category and moves them to the new ObjMgrGenericInfo category such as the current Workspace Version used by the object manager.

What's New in Siebel System Monitoring and Diagnostics Guide, Siebel CRM 23.6 Update

The following information lists the changes in this revision of the documentation to support this release of the software.

What's New in Siebel System Monitoring and Diagnostics Guide, Siebel CRM 23.6 Update

Topic	Description
Enable the Logging for the AI and for the Target Server Component *Review the Log Files for the Details*	New topics: These topics are a part of *Configuring Logging and Monitoring for Siebel Application Interface* They depict the steps you need to perform to generate detailed logging, to capture requests from AI to Gateway to the Siebel server component and task.

What's New in Siebel System Monitoring and Diagnostics Guide, Siebel CRM 22.4 Update

The following information lists the changes in this revision of the documentation to support this release of the software.

What's New in Siebel System Monitoring and Diagnostics Guide, Siebel CRM 22.4 Update

Topic	Description
Configuring the Common Logger for Siebel Application Interface *Configuring Siebel Gateway Log Files*	Modified topics. You can configure the log4j2-siebel.properties file to include the **`monitorInterval`** parameter. This parameter allows changing the log level at runtime for all application container logs that are controlled by this file and not by profiles in Siebel Management Console.

What's New in Siebel System Monitoring and Diagnostics Guide, Siebel CRM 21.6 Update

The following information lists the changes in this revision of the documentation to support this release of the software.

What's New in Siebel System Monitoring and Diagnostics Guide, Siebel CRM 21.6 Update

Topic	Description
Configuring the Common Logger for Siebel Application Interface	Moved topic here from *Siebel Installation Guide* .
Configuring Siebel Gateway Log Files	Modified topic. As of Siebel CRM 18.9 Update, logging for the cloudgateway.log file now involves editing the log4j2-siebel.properties file, and no longer uses values from the gateway.properties file.

What's New in Siebel System Monitoring and Diagnostics Guide, Siebel CRM 21.3 Update

The following information lists the changes in this revision of the documentation to support this release of the software.

What's New in Siebel System Monitoring and Diagnostics Guide, Siebel CRM 21.3 Update

Topic	Description
Configuring Siebel Gateway Log Files	Modified topic. As of Siebel CRM 21.2 Update, the **`applicationcontainer`** directory has been replaced by two directories, as follows: • **`applicationcontainer_external`** (for Siebel Application Interface) • **`applicationcontainer_internal`** (for all other Siebel Enterprise components) For more information, see *Siebel Installation Guide* .

2 Configuring Logging and Monitoring for Siebel Application Interface

Configuring Logging and Monitoring for Siebel Application Interface

This chapter describes configuring the Siebel Application Interface for logging and monitoring purposes. It includes the following topics:

About Logging for Siebel Application Interface

Siebel Application Interface writes several different types of logs. These logs include the following:

- UI.log for user applications
- EAI.log for EAI applications
- RESTInBoundDefault.log to capture details for Inbound REST requests for EAI applications
- siebel_1.log to capture details for all inbound traffic through the AI layer for all channels from Siebel 20.8 and above
- CommonLoggerLog.log for some common events like startup and shutdown of Siebel Application Interface

The logs noted here are written in the `SiebelAI_Root\log` directory. Standard application container logs are created in the same directory.

Configuring Siebel Application Interface Logging

For Siebel Application Interface, log levels are configured in the Application Interface profile. For more information, see the chapter about configuring Siebel CRM server modules in *Siebel Installation Guide* .

Configuring the Common Logger for Siebel Application Interface

You can configure the log level for the Siebel Application Interface common logger, which is used for certain common logging tasks. This log cannot be configured using Siebel Management Console, unlike other types of logging for Siebel Application Interface described in *Siebel Installation Guide* .

To configure the log level for the common logger, you edit the file log4j2-siebel.properties, which is located in the directory `SIEBEL_AI_ROOT\applicationcontainer_external\webapps\siebel\WEB-INF`.

When configuring the log level for the common logger, modify only the following line, and no other part of the file. For example, you can change the level from ERROR to INFO. You can use the following values: TRACE, DEBUG, INFO, WARN, ERROR, or FATAL. The default log level configured in the file is ERROR.

```
logger.CommonLogger.level = ERROR
```

> **Note:** Settings such as INFO, DEBUG, or TRACE can exponentially increase the size of the log files, consume a large amount of disk space, and reduce performance of the Siebel Application Interface node. Use these settings only when some investigation is needed.

If you plan to configure the log level for the common logger, it is recommended to do so after you finish configuring Siebel Application Interface using Siebel Management Console. After modifying the log4j2-siebel.properties file, you must restart the application container, as described in *Siebel Installation Guide* .

Optionally, you can configure the log4j2-siebel.properties file to include the `monitorInterval` parameter, which allows changing the log level at runtime for all application container logs that are controlled by this file and not by profiles in Siebel Management Console. If you change the following line:

```
status = warn
name = Siebel
monitorinterval = 300
```

Then Apache Tomcat checks for changes in the log level within the specified interval (300 seconds, or 5 minutes) and, if the log level changed, then the log statements are written to log files to reflect the change in the log level.

Related Topics

Configuring Siebel Gateway Log Files

Related Books

Siebel Installation Guide

About Siebel Application Interface Monitoring

Monitor the Siebel Application Interface by configuring and reading the Siebel Application Interface statistics page. This HTML page provides current information about the operations and communications of the Siebel Application Interface, which allows system administrators to have a better understanding of the use of this module. Each of the sections of the statistics page lists measurable objects, their values, mean values, and standard deviations.

> **CAUTION:** As the Siebel Application Interface statistics page provides sensitive information about the type of requests running and potentially active sessions, it is strongly recommended that this page be protected with an authentication mechanism, such as from a third-party provider.

Configuring the Siebel Application Interface Statistics Page

The Siebel Application Interface statistics page is configured when you configure applications on the Siebel Application Interface using Siebel Management Console. This page is located at the following URL:

https://ApplicationInterfaceHost:Port/siebel/stats

For more information about specifying the Siebel Application Interface statistics page and related settings when you configure the Siebel Application Interface profile, see *Siebel Installation Guide* .

You can specify whether statistics are gathered on current sessions and then reported to the statistics page. If Monitor Sessions is enabled, then, when sessions are created, they are entered into the statistical repository and appear on the application's Siebel Application Interface statistics page. This setting allows system administrators to determine who is logged onto the system at any given time, and to determine the session ID with a given user in a non-debug log level. However, performance is slightly degraded by using this feature. If Monitor Sessions is not enabled, then sessions are not monitored by the statistical repository and do not appear in an application's statistics page.

Accessing the Siebel Application Interface Statistics Page

The Siebel Application Interface statistics page is generated by the Siebel Application Interface. To access the Siebel Application Interface statistics page, enter a URL like the following in a Web browser:

https://ApplicationInterfaceHost:Port/siebel/stats

When accessing the Siebel Application Interface statistics page URL, additional parameters can be appended to the URL, which modify the display and content of the page.

Statistics Page Verbosity Option. This option allows the user to dictate the amount of information that is to appear in Siebel Application Interface statistics page. There are three settings as shown in the following table.

Verbose Parameter Setting	Description
Verbose=Low	Default value if not present. Displays only system and application-level statistics.
Verbose=Medium	Displays the low-setting information, plus the lock statistics.
Verbose=High	Displays the medium-setting information, plus currently active operations to the Siebel Server.

Statistics Page Reset Option. This option allows the user to dictate whether the statistics are reset after viewing. There are two settings as shown in the following table.

Verbose Parameter Setting	Description
Reset=True	Resets noncounter and current operational statistics.
Reset=False	Default value if not present. Does not reset current operational statistics.

Examples of the Siebel Application Interface statistics page request with parameters:

- https://ApplicationInterfaceHost:Port/siebel/stats?Verbose=High&Reset=True

  This request displays the System Stats, Applications, Current Sessions, Locks, and Current Operations Processing statistical categories and then resets noncounter and current operational statistics.
- https://ApplicationInterfaceHost:Port/siebel/stats?Reset=True

  This request displays the System Stats and Applications statistical categories and then resets noncounter and current operations statistics.

Events and Objects on the Siebel Application Interface Statistics Page

The individual events and objects measured on the Siebel Application Interface statistics page are described in the following list. For examples of these metrics, see *Example of Siebel Application Interface Statistics Page*.

- **Open Session Time.** This event reflects the total amount of time it took to open a session. In the general stats section, the count is the number of times a session was opened and the mean reflects the average time it took to open a session.

- **Response Time (waiting for service event).** This event measures the time it takes to receive a callback response from the Siebel server. This event functions with CTI and internal login callbacks. A callback is a mechanism used by the Siebel Server to initiate communication with the Siebel Application Interface.
- **Close Session Time.** 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.
- **Request Time (waiting for service method to process).** This event is the amount of time it takes to submit a request to the Siebel Server and to get a response back. For example, if the user (on the browser) clicked on a button, then the Siebel Application Interface receives the request and invokes a service on the Siebel Server. The value for Request Time is the total amount of time for invoking that service.
- **Applications.** This section displays information about the various applications, for example, session life span and number of attempts to use the application.
- **Current Sessions.** This section contains information about the current active sessions open. The parameter SessionMonitor must be set to True for this to take effect (see *Configuring the Siebel Application Interface Statistics Page* for more information about SessionMonitor). If verbose mode is used, then this section also displays the anonymous sessions (see *Accessing the Siebel Application Interface Statistics Page* for more information about verbose mode).
- **Current Operations Processing.** Use the information in the following table when troubleshooting a process that might have stopped responding.

  The Current Operations Processing section contains a table that shows current requests that are in progress. The following table shows the operations that are running and the duration of each operation (in seconds). Some of these requests have been running for more than 10 seconds. A request with a large duration value might not be responding. If a request never completes, then it has effectively stopped responding.

Operation	Duration
<server>://172.20.232.19:2320/siebel/SCCObjMgr/!7.fb8.ddde. <snip>	3888.0282
<server>://172.20.232.19:2320/siebel/SCCObjMgr/!8.f54.df75.3c07ef90 <snip>	20.8209
<server>://172.20.232.19:2320/siebel/SCCObjMgr/!8.f54.df75.3c07ef90 <snip>	0.2796

The following table provides descriptions of some of the Siebel Application Interface statistics and is followed with the definition of those statistics.

	Type of Statistic		
Statistic	**General**	**Frequency**	**Description**
Count	Yes	No	Accumulative count of the events
Mean	Yes	No	Average value for the event

	Type of Statistic		
Statistic	**General**	**Frequency**	**Description**
	No	Yes	Average period for which one such event occurs (in seconds)
Standard deviation	Yes	No	Standard deviation for the event
	No	Yes	Standard deviation (in seconds)

Statistics results can vary depending on the session type. For example, for an anonymous session requested from the pool event type, the statistics provide the following information:

- **General statistics count.** The accumulative count of the anonymous session requests.
- **General statistics mean.** The average value for anonymous session requests (the value is 1 or greater).
- **General statistics standard deviation.** The standard deviation for anonymous session requests (the value is zero [0] or greater).
- **Frequency mean.** The average period between anonymous session requests in seconds.
- **Frequency stddev.** The standard deviation in seconds.

However, for an open session time event type, the statistics provide the following information:

- **General statistics count.** The accumulative open session times.
- **General statistics mean.** The average value for an open session time.
- **General statistics standard deviation.** The standard deviation for an open session time.
- **Frequency mean.** The average period between open session events in seconds.
- **Frequency standard deviation.** The standard deviation in seconds.

The next topic, *Example of Siebel Application Interface Statistics Page*, provides examples that show how these statistics might occur with different types of events.

Example of Siebel Application Interface Statistics Page

A sample Siebel Application Interface statistics page is reproduced in the tables in this topic. The information contained in these tables encompasses one Siebel Application Interface statistics page.

The following table shows sample system statistics.

Event	Value (Seconds)	General Statistics (Count, Mean, Standard Deviation)	Frequency (Mean, Standard Deviation)
Open Session Time	191.6682	12 15.9723 34.4210	61.9689 128.9318

Event	Value (Seconds)	General Statistics (Count, Mean, Standard Deviation)	Frequency (Mean, Standard Deviation)
Response Time (waiting for service event)	0.0000	0 0.0000 0.0000	0.0000 0.0000
Close Session Time	0.0000	0 0.0000 0.0000	0.0000 0.0000
Request Time (waiting for service method to process)	349.9513	23 15.2153 70.4652	3374.4503 16020.5422

The following table shows sample current sessions.

Event	Total Time (Seconds)	General Statistics (Count, Mean, Standard Deviation)	Frequency (Mean, Standard Deviation)
siebel://test:2320/siebel/objmgr/test/!1.64c.14.3bb0e99fuser0	3.9228	4 0.9807 0.8953	85.9297 168.6426
siebel://test:2320/siebel/objmgr/test/!9.34b.1fe.3bbf349fuser1	338.4631	9 37.6070 112.8092	59.4458 116.0594
siebel://test:2320/siebel/objmgr/test/!1.56.1ef.4c0a0e99fuser2	3.3424	3 1.1141 0.8227	25665.0354 44450.4096

The following table shows a sample current operations processing.

Operation	Duration (Seconds)
NewAnonSession_00000022_499	0.9581

Operation	Duration (Seconds)
Open Session Time_00000023_499	0.9580

Enable the Logging for the AI and for the Target Server Component

This topic explains the steps you need to perform to generate detailed logging, to capture requests from AI to Gateway to the Siebel server component and task:

1. Set `SMC Logging = Information`

   Log into SMC > Profiles > Application Interface > select the application profile being used > Logging section > set everything to INFORMATION level.

   INFORMATION level is sufficient for capturing the details. If desired, it is also possible to set these to DEBUG level as well.

   > **Note:** Both Information or Debug level will work fine.

2. Edit the `<AI>\applicationcontainer_external\webapps\siebel\WEB-INF\log4j2-siebel.properties` set to ALL for the CommonLogger.

   ```
   <logger name="CommonLogger" level="ALL" additivity="false">
   <AppenderRef ref="CommonLogger"/>
   </logger>
   ```

3. Edit the `<AI>\applicationcontainer_external\conf\server.xml` bottom AccessLogValve section to have more descriptive information/labels for the log entry output.

   > **Note:** Please refer to the Tomcat documentation for additional information on the flags within the AccessLogValve: *https://tomcat.apache.org/tomcat-9.0-doc/config/valve.html*

Flag and Pattern	Tomcat Loggings Flags Description and Usage
%{begin:yyyy-MM-dd HH:mm:ss.SSS}t	Request start time(formatted with label "Begin Time:" and yyyy-MM-dd HH:mm:ss.SSS)
%{end:yyyy-MM-dd HH:mm:ss.SSS}t	Response finish time (formatted with label "End Time:" and yyyy-MM-dd HH:mm:ss.SSS)
%r	First line of the request (method and request URI)

Flag and Pattern	Tomcat Loggings Flags Description and Usage
%F	Time taken to commit the response, in milliseconds
%s	HTTP status code of the response with prefix label 'http'
%{Content-Length}i	Write value of incoming header with name Content-Length
%b	Bytes sent, excluding HTTP headers, or '-' if zero
%{SOAPACTION}i	Write value of incoming header with name SOAPACTION

> **Note:** Adding the extra strings/labels such as PROCESSING TIME, HTTPS, MS, BYTES will help when reviewing the logs, provides more description to what the numerical values represent and easier when parsing/searching for patterns in the log file.

4. Restart the AI container service after making the above changes.
5. Set logging on the target Siebel server component.

   For instance the EAI OM component for inbound interfaces such as Inbound Web Services, Inbound REST, etc.

   a. Set all events on the EAI OM component to `log level = 5` (preferred, in order to generate the maximum detail possible).
   b. Set the OM component parameter `Enable Business Service Argument Tracing = TRUE"`

      This will generate the `Inbound_*.dmp file` with the incoming soap payload data and SessionToken string, which then can be used to match back to the EAI OM task log file by matching the ending file's TaskId value.
   c. Restart the EAI OM component afterwards as changing the **Enable Business Service Argument Tracing** parameter requires a component restart.

Review the Log Files for the Details

After setting the logging and restart the AI service and Server component, log into the Siebel Client UI or send an inbound request to the EAI OM component.

Then, check for these files in AI log directory after using the above logging setting:

1. `<AI>\applicationcontainer_external\logs\CommonLogger.log` : This file shows the gateway returning the ConnectString to the server/OM component that the request will be sent to.

2. `<AI>\applicationcontainer_external\logs\localhost_access_log.<DATE>.txt` : This `localhost_access_log.<DATE>.txt` shows the request that came through the AI instance.

   The first field shows the IP Address of where the request came from (%h - Remote hostname (or IP address if the resolveHosts attribute is set to false; by default the value is false)).

   > **Note:** Recall the server.xml file AccessLogValve section was modified to include additional text so it now shows more description for the time and http status code with the following:

   ```
   <Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs"
   prefix="localhost_access_log" suffix=".txt"
   pattern="%h %l %u Begin Time: %{begin:yyyy-MM-dd HH:mm:ss.SSS}t, End Time: %{end:yyyy-MM-dd
    HH:mm:ss.SSS}t &quot;%r&quot; Response Commit Time %F ms, Request Processing Time %D ms,
   http %s, IContentLength %{Content-Length}i bytes, OContentLength %b bytes, %{SOAPACTION}i" />
   ```

   > **Note:** Recall the server.xml file AccessLogValve section was modified to include additional text so it now shows more description for the time, http status code, request begin and end times, etc.

3. `<AI>\applicationcontainer_external\logs\siebel_1.log` :

   The siebel_1.log shows which Siebel server Host machine name, Component OM PID Value and TaskId the request is routed to.

   > **Note:** This [HOST/PID/TaskId] information in the siebel_1.log file is only seen for Inbound Web Services using Session Management and for Inbound REST Connections.

   > **Note:** The [HOST/PID/TaskId] does not show up in the siebel_1.log file for Inbound Web Service Anonymous Pool connections, for more information see, [Inbound Web Service Anonymous Pool] .

4. For EAI OM or any Siebel OM component:

   Check the Siebel server `siebsrvr\log` directory on the server HOSTNAME shown from the above siebel_1.log file:

   a. Check for the Object Manager log for the component. Example `EAIObjMgr_enu_*.log` for EAI OM component, or `sccobjMgr_enu_*.log` for Call Center ENU

      In particular, locate the OM log file with the ending TaskId which matches what is shown in the above `siebel_1.log` file.

      Open the OM log file to inspect the first line to confirm it matches the ProcessId and TaskId value from the `siebel_1.log` file.

   b. If the calls are for Inbound Web Services, check for `Inbound_*.dmp` file on the same Siebel server host machine where the EAI OM task log files are found.

      These Inbound_*.dmp files come in pairs:

      - **InboundDispatcher_input_args_<TaskId>.dmp** - This contains the inbound SOAP request received by Siebel.
      - **InboundDispatcher_output_args_<TaskId>.dmp**- This contains the SOAP response sent out from Siebel. The SOAP response may be a SOAP fault message if your inbound request was not valid.

      The ending TaskId value of these Inbound_*.dmp file should match the ending TaskId of an EAIObjMgr_enu log file as well. This means, that a particular EAIObjMgr_enu task processed that particular inbound message captured in the Inbound_*.dmp file with the same TaskId value.

      If the Web Service uses Session Management and SessionToken, then one would also see these .dmp file followed by numbers such as (1), (2), (3) .... (N) with the same TaskId, which means that the same EAIObjMgr_enu task was opened and remained open to process these requests coming in with SessionToken values. This is one way to determine how many inbound web service requests were processed by a particular EAIObjMgr_enu task.

      Within the Inbound_*.dmp file, there are also actual payload and SessionToken, this will allow an administrator to find out which Siebel server, component, task that the particular request was routed to and handled by.

3 Monitoring Siebel Server Run-Time Operations

Monitoring Siebel Server Run-Time Operations

Monitoring Oracle's Siebel Server run-time operations is a necessary, on-going aspect of administering a Siebel application. Use metrics such as log files, state values, and statistics to monitor the Siebel application performance.

This chapter includes the following topics:

About Siebel Server States

After installation and initial configuration using Siebel Management Console, a Siebel Server is always in one of the following states when connected to the Server Manager component (alias ServerMgr):

- **Starting Up.** Indicates that the Siebel Server is in the process of starting up. When this process is complete, the state changes to Running.
- **Running.** Indicates that the Siebel Server is running and that Siebel Server components can operate. This is the normal mode of operation for the Siebel Server. When the Siebel Server Service starts, it sets the Siebel Server to the Running state by default (depending on the value of the Auto Startup Mode Siebel Server-level parameter, which defaults to TRUE).

  When the Siebel Server starts, its components are enabled and the default number of tasks is instantiated for the background mode components (the number of tasks is determined by the value of the Default Tasks parameter for each component).
- **Shutting Down.** Indicates that the Siebel Server is in the process of shutting down. When this process is complete, the state changes to Shutdown.
- **Shutdown.** Indicates that the Siebel Server is running, but component tasks are not currently running (other than the Siebel Server Manager component, which is operational whenever the Server Manager is connected) and new tasks are not allowed to start. The only processes that can run when the Siebel Server is in a Shutdown state are the Siebel Server System Service itself and the Server Manager for a Siebel Server Manager client.

  Shut down the Siebel Server using the Server Manager whenever you want to shut down the:

  - Server computer on which the Siebel Server is running. This allows a clean shutdown of each Siebel Server component.
  - Siebel Server to perform maintenance.
  - Siebel Server to perform an automatic upgrade on the Siebel Server's software using Siebel Upgrade Wizard.

  > **Note:** Individual components might be shut down or disabled without having to shut down the entire Siebel Server.

If the Siebel Server is not connected to the Server Manager component (alias ServerMgr), then the following states are applicable:

- **Not Online.** The server component is not online. After the Siebel Server is restarted, this component state might occur temporarily before the component's state becomes Online. If the status Not Online persists, then an error is preventing the component from becoming online. Check the component log and fix the error to let the component state become Online again.
- **Partially Offline.** The server component is partially offline and cannot start until the Siebel Server is restarted.

  - For a multithreaded component, if the number of active running processes is less than the value of the parameter MinMTServers, then the state is Partially Offline.
  - For a background mode component, if the number of active running processes is less than the value of the parameter DfltTasks, then the state is Partially Offline.

- **Not available.** Indicates that the Siebel Server has not been started. Indicates that the Server Manager cannot connect to the Siebel Server; you cannot run any tasks or perform any administrative functions on that Siebel Server.
- **Connect Failed.** Indicates that Server Manager is able to get the connect string for the ServerMgr component from the Siebel Gateway but is unable to connect to the Siebel Server.
- **Handshake Failed.** On startup, Server Manager sends a handshake request to the Siebel Server for the ServerMgr component. If that request fails, then this state occurs. Also, if the ServerMgr component on that particular Siebel Server cannot start any more tasks (because it has reached Maximum Tasks (alias MaxTasks) number of tasks) for the administration clients, then this state occurs. For more information about the MaxTasks parameter, see *Siebel System Administration Guide* and *Siebel Performance Tuning Guide* .
- **Login Failed.** Server Manager connects to every Siebel Server for authentication. If the authentication fails for any Siebel Server, then the Login Failed state appears.
- **Disconnected.** When Server Manager connects to the Siebel Server, the Siebel Server starts a task for the ServerMgr component. If that task exits (because of a malfunction or other problems), then the Disconnected state appears.

Siebel Server Status Fields

Each Siebel Server record has three fields in which the Siebel Server status appears. The following table provides the Siebel Server status fields.

GUI Column Name	Command-Line Interface Column Name	Description
Server State (Internal)	SBLSRVR_STATE	The state of the Siebel Server.
State	SV_DISP_STATE	The state of the Siebel Server using the appropriate language code.
State (Icon)	Not applicable	A stoplight icon representing the state of the Siebel Server. Clicking the icon in the State field reveals the actual state value associated with the stoplight color, which can be one of the following for example: • Running (Green stoplight), which indicates normal conditions. • Completed (Green stoplight), which indicates normal conditions. • Exited (Red stoplight), which indicates a non-operational condition. **Note:** The State (Icon) field is blank when you are not connected to a Siebel Server.

About Siebel Server Component Group States

A component group might be in one of several states. The run state is dependent on the enable state; only component groups that have an Online enable state when the Siebel Server was started can have a run state of Online or Running:

- **Online.** Every component within the component group is enabled to run tasks.
- **Running.** Every component within the component group is enabled, and at least one component within the component group is running a task.
- **Shutdown.** Every component within the component group is shut down. Tasks cannot run for any components within the component group.
- **Part shutdown.** At least one component within the component group is shut down or shutting down.
- **Offline.** Every component within the component group is offline.
- **Part offline.** At least one component within the component group is offline or unavailable.
- **Starting up.** At least one component within the component group is starting up.

Server Component Group Status Fields

Each Siebel Server component group record has three fields in which the status appears as shown in the following table.

GUI Column Name	Command-Line Interface Column Name	Description
State	CA_RUN_STATE	The state of the server component group using ENU language code.
Run State (internal)	CA_RUN_STATE	The state of the server component group using the appropriate language code.
State (Icon)	Not applicable	A stoplight icon representing the state of the Siebel Server component group. Clicking the icon in the State field reveals the actual state value associated with the stoplight color, which can be one of the following, for example: • Running or Online (Green stoplight), which indicates normal conditions. • Part Offline (Yellow stoplight), which indicates a temporary non-operational condition. • Shutdown or Offline (Red stoplight), which indicates a non-operational condition.

About Siebel Server Component States

A Siebel Server component might be in one of the following states: Starting Up, Online, Running, Offline, Shutting Down, Shutdown, or Unavailable.

The Siebel Server component state is dependent on the assignment state of the component group to which it belongs; only Siebel Server components within assigned component groups when the Siebel Server was started can be Running or Online:

- **Starting Up.** Indicates that the Siebel Server component is in the process of starting up. When this process is complete, the state changes to Online. When a new task is started for the component, the component state changes to Starting Up during the initialization phase and then to Running.
- **Online.** Indicates that tasks are currently not running for the Siebel Server component, but new tasks might be started through the Siebel Server Manager (or in response to client requests, for interactive-mode components). When the Siebel Server starts, components for which processes are *not* started by default are online.
- **Running.** Indicates that tasks are currently running for the Siebel Server component on the Siebel Server, and new tasks are allowed to start (up to the value of the Maximum Tasks parameter for the component). When the Siebel Server starts up, background-mode components for which processes are started by default (components with a Default Tasks parameter set to a nonzero value) start.
- **Offline.** Indicates that new tasks might not be started for the component, though current running tasks can continue running (for background-mode components) or run to completion (for batch-mode and interactive-mode components).

  You might want to disable an individual component to perform a system maintenance operation outside of the Siebel Server. For example, you might disable the Synchronization Manager component to do a file system reorganization on the docking subdirectory.

  To minimize the number of multithreaded processes started on the Siebel Server, you might want to disable components that you do not plan to run.

  You might also want to disable components due to database licenses. If you have exceeded the maximum licensed connections for your database, then you might want to disable the Siebel Server components that you plan not to use. You must only disable components for which you do not plan to run tasks across the entire enterprise. Setting the Min MT Servers parameter to 0 for multithreaded Siebel Server components renders the server component unable to run tasks.

  An offline component might be set to Online (or Started, if there are still tasks running for the offline component) or Shutdown, in which case, any running tasks are stopped as cleanly as possible.
- **Shutting Down.** Indicates that the Siebel Server component is in the process of shutting down. When this process is complete, the state changes to Shutdown.
- **Shutdown.** Indicates that processes are not running for the component and new tasks might not be started. Each task running when the component shuts down is stopped as soon as possible. Components are set to Shutdown when the Siebel Server shuts down, with the exception of the Siebel Server Manager component, which remains Online to perform administrative commands executed by the Siebel Server Manager. Background-mode components that are set to Shutdown but have a Default Tasks parameter set to a nonzero value might be set to Online or Started.
- **Unavailable.** Indicates that processes are not running for the component when a Siebel Server process is running. Multithreaded Siebel Server components change to an Unavailable component state when the Min MT Servers parameter is set to a value greater than 0 and no Siebel Server processes are actually running for that component. In this case, the Siebel Server component might exit with an error and become unavailable because it failed to initialize. Siebel Server components might also go into this state if the database connection is down. In this case, you must restart the Siebel Server component after the database connection has been reestablished.

Server Component Status Fields

Each server component record has two fields in which the status appears as shown in the following table.

GUI Column Name	Command-Line Interface Column Name	Description
State	CP_DISP_RUN_STATE	The state of the Siebel Server component using the appropriate language code.
State (Icon)	Not applicable	A stoplight icon representing the state of the Siebel Server component. Clicking the icon in the State field reveals the actual state value associated with the stoplight color, which can be one of the following for example: • Running or Online (Green stoplight), which indicates normal conditions. • Shutting Down (Yellow stoplight), which indicates a temporary non-operational condition. • Shutdown or Offline (Red stoplight), which indicates a non-operational condition.

About Siebel Server Task States

A Siebel Server task is an instantiation of a Siebel Server component. To run a Siebel Server task, you must run a component job, which requests one or more Siebel Server tasks to run. For information about component jobs, see *Siebel System Administration Guide* .

A Siebel Server task might be in one of four fundamental states: Running, Paused, Stopping, or Completed.

- **Running.** Indicates that the task is executing normally. While the task is running, it periodically updates its task status, a component-generated message that indicates the task progress (or phase of operation).
    - Background mode component tasks run until stopped manually, or until the Siebel Server or the server component shuts down.
    - Batch mode component tasks run to completion when their assigned unit of work is done.
    - Interactive mode component tasks run until the client signs off from the connection (or until the task, server component, or Siebel Server is shut down).

      You might explicitly stop any currently running component task.
- **Paused.** Indicates that the task has been temporarily placed in a suspended state. A paused task does not exclusively hold any shared system resources (such as file locks or database locks), or expend any processor or I/O cycles. You might choose to pause a running task to temporarily free up the system to process other critical tasks without having to restart the entire task. You might then resume or stop the paused task.

  > **Note:** Only tasks from certain component types can be paused. For a list of these component types, see *Siebel System Administration Guide* .

- **Stopping.** Indicates that the task has been instructed to stop, or the server component or Siebel Server is being shut down. Occasionally, the shutdown process might take a while, in which case you might issue another Stop command, and the shutdown is forced (this state might appear as Forcing Shutdown). After a task has been instructed to stop, it might not be resumed.
- **Completed.** Indicates that the task is no longer running. After a task is completed, it might not be restarted, though you might start a new task for the same server component. Several variations exist for the Completed state, depending on the manner in which the task finished processing:
  - *Completed* indicates that the task ran to completion and exited normally (batch mode and interactive mode tasks only).
  - *Exited with Error* indicates that the task encountered an error during its processing (such as bad input values or database errors). In this case, the Task Status field displays the error identifier for the error that has occurred.
  - *Killed* indicates that the task was not able to shut down cleanly, and you forced the task to shut down.

About Task Status Fields

Each Siebel Server record has three fields in which the Siebel Server status appears. The following table provides the task status fields.

GUI Column Name	Command-Line Interface Column Name	Description
State	TK_RUNSTATE	The state of the task using the appropriate language code.
Status	TK_STATUS	Every component task sets various state values during the course of its operation. The Status column in the tasks view and the TK_STATUS column in the command-line interface displays the value for the state value Task Status (alias TaskStatus).
State (Icon)	Not applicable	A stoplight icon representing the state of the Siebel Server task. Clicking the icon in the State field reveals the actual state value associated with the stoplight color, which can be one of the following, for example: • Running or Completed (Green stoplight), which indicates normal conditions. • Paused (Yellow stoplight), which indicates a temporary or non-operational condition. • Exited or Exited with Error (Red stoplight), which indicates a non-operational condition.

About Component Job States

After the creation of a component job, it is always in one of the states in the following list. For more information about starting component jobs, see *Siebel System Administration Guide* . For more information about monitoring component job status, see *Monitoring Component Job Status*.

- **Creating.** Indicates the component job record is in the process of being defined.

- **Queued.** Indicates the component job record was started and is scheduled to run. The component job field Scheduled Start defines when the component job runs.
- **Active.** Indicates the scheduled component job is running.
- **On Hold.** Indicates the component job is on hold and will not run at the Scheduled Start time. Only component jobs in the queued state can be put on hold.
- **Cancelled.** Indicates the component job is cancelled. Only component jobs in the queued or on hold state can be cancelled.
- **Canceling.** Indicates the component job is in the process of being cancelled.
- **Error.** Indicates the component job ran, but encountered an error during operation.
- **Success**. Indicates the component job ran and completed successfully.
- **Completed.** Indicates that repeating component jobs completed successfully.
- **Expired.** Indicates the component job has expired. The component job field Expiration Date defines when the component job expires.
- **Parent Request Cancelled.** Indicates the first component job of a repeating component job was cancelled. The first component job of a repeating component job is considered the parent job.
- **Parent Request On Hold.** Indicates the first component job of a repeating component job is on hold. The first component job of a repeating component job is considered the parent job.

About User Sessions

User sessions include data on any user logged into the Siebel Server as well as sessions created by the Siebel application. User sessions comprise all interactive component tasks.

User sessions run based on a Siebel Server component task. Therefore, user sessions have the properties of Siebel Server component tasks. The Session ID field of an individual user session shares the same ID number as the Task ID of the component task that runs the session. That is, information about user sessions can be viewed as either a user session or a task.

For information and procedures on monitoring user sessions, see *Monitoring User Session Status*. For information and procedures on monitoring tasks, see *Monitoring Server Component Task Status*.

About Siebel Application Statistics

Various statistics are recorded at the task level for every Siebel Server component task. You might use these statistics to:

- Monitor the progress and performance of a task, component, or Siebel Server
- Optimize system performance

When the task completes its operation, task-level statistics (gathered dynamically during the operation of a task) roll up to the component and Siebel Server levels.

Two types of statistics exist for task-level Siebel Server statistics:

- **Subsystem statistics.** Common to every component process (such as process management, networking, database access, and file I/O) and tracked for each component task.

- **Component-specific statistics.** Only applicable to the component for which the statistics are defined.

When a task for a component completes its operation, both generic and component-specific statistics roll up to the component level. Only generic statistics roll up to the Siebel Server level.

Statistics on the component level includes data for completed tasks on interactive and batch mode components. Statistics for component tasks that are still running are not included. Check the tasks directly to monitor statistics for running tasks on interactive and batch mode components. For information about monitoring task statistics, see *Monitoring Server Component Task Statistics*. For background mode components, the statistic rollup behavior is slightly different because the component tasks are never complete. For background components, the component statistics change whenever a statistic value is updated by the running component task. For a listing and brief descriptions of Siebel application statistics, see *List of Statistics and State Values*.

> **Note:** If some Siebel application statistics are not visible, then set the Show Advanced Objects (alias ShowAdvancedObjects) parameter to TRUE for the server component Server Manager (alias ServerMgr). For more information about advanced objects, see *Siebel System Administration Guide* .

About Siebel Application State Values

State values contain information about the current operation of a task or the component for which the task is running. Component tasks periodically update their state values to indicate information about their current processing, such as the current phase of operation. State values are defined at the component and task levels. Component-level state values refer to the state of the component as a whole. Task-level state values refer to the state of an individual process for a Siebel Server component.

Two types of state values exist for components and component tasks:

- **Subsystem state values.** Kept for every component (such as Component Start Time and Component Stop Time) and component task (such as Task Start Time and Task Stop Time) that uses that subsystem.
- **Component-specific state values.** Kept for every component and component task. Only applicable to the component for which they are defined.

Monitoring Siebel Enterprise Server Status

Monitor the status of Siebel Servers in a Siebel Enterprise Server by using the Server Manager GUI or the Server Manager command-line interface program (srvrmgr). For configuration tasks and background information about the Siebel Enterprise Server, see *Siebel System Administration Guide* .

To monitor a Siebel Enterprise Server using the Server Manager GUI

- Navigate to the Administration - Server Management screen, Enterprises, and then the Servers view.
  The following information appears:
  - The name and description of the Siebel Enterprise Servers available are in the Enterprise Servers list.
  - The state of the Siebel Servers for the selected Siebel Enterprise Server are available in the Servers list. For details on Siebel Server states, see *About Siebel Server States*.

- The state of the Siebel Server components for the selected Siebel Server are available in the Components list. For details on Siebel Server component states, see *About Siebel Server Component States*.

To monitor Siebel Enterprise Server using srvrmgr

- At the srvrmgr program prompt, enter:

```
list servers
```

CAUTION: Make sure you do not start the Server Manager command-line interface program for a particular Siebel Server; that is, do not start the Server Manager command-line interface with the `/s` flag.

For details on starting, running, and configuring the Server Manager command-line interface program, see *Siebel System Administration Guide* .

Monitoring Siebel Server Status

Monitor the status of Siebel Servers by using the Server Manager GUI or Server Manager command-line interface program (srvrmgr program). The following topics describe procedures that monitor the Siebel Server:

- *Monitoring Siebel Server State*
- *Monitoring Siebel Server Component Groups*
- *Monitoring Siebel Server Log Files*
- *Monitoring Siebel Server Statistics*
- *Monitoring Siebel Server Tasks*
- *Monitoring Siebel Server Tasks*

For background information about Siebel Servers, including running and configuring procedures, see *Siebel System Administration Guide* .

Monitoring Siebel Server State

Monitor the status of a Siebel Server by using the Server Manager GUI or the Server Manager command-line interface program (srvrmgr). For details on the possible states of the Siebel Server, see *About Siebel Server States*. For information about monitoring other Siebel Server run-time operations, see *Monitoring Siebel Server Status*.

To monitor the Siebel Server state using the Server Manager GUI

1. Navigate to the Administration - Server Management screen, then Servers view.
2. In the Servers list, select the Siebel Server of interest.
3. Review the state of the selected Siebel Server by viewing the State (Icon) or Server State fields.

To monitor the Siebel Server state using srvrmgr

- At the srvrmgr program prompt, enter:

```
list servers
```

For details on starting, running, and configuring the srvrmgr program, see *Siebel System Administration Guide* .

Monitoring Siebel Server Component Groups

Monitor the status of component groups for a Siebel Server using the Server Manager GUI or the Server Manager command-line interface program (srvrmgr). For details on Siebel Server component group states, see *About Siebel Server Component Group States*. For information about monitoring other Siebel Server run-time operations, see *Monitoring Siebel Server Status*.

To monitor component groups using Server Manager GUI

1. Navigate to the Administration - Server Management screen, then Servers view.
2. In the Servers list, select the Siebel Server of interest.
3. From the view tabs, click Component Groups.
4. Review the state of the component groups for the selected Siebel Server by viewing the State (Icon) and State fields of each component group record.

To monitor component groups on srvrmgr

- At the srvrmgr program prompt, enter

```
list component groups for server siebel_server_name
```

For details on starting, running, and configuring the srvrmgr program, see *Siebel System Administration Guide* .

Monitoring Siebel Server Log Files

Monitor the log files for a Siebel Server using the Server Manager GUI. You can also review Siebel Server log files by manually accessing the file or querying the file with the Log File Analyzer (LFA) utility.

For background information about:

- Siebel Server log files, see *About Siebel Server Log Files*.
- LFA, see *About the Log File Analyzer*.
- Event logging, see *About Configuring Siebel Server and Component Logging*.

For information about monitoring other Siebel Server run-time operations, see *Monitoring Siebel Server Status*.

To monitor Siebel Server log files on Server Manager GUI

1. Navigate to the Administration - Server Management screen, then Servers view.

2. In the Servers list, select the Siebel Server of interest.
3. From the view tabs, click Log.

   Each entry in the Log view list represents an event logged in the Siebel Server log file. For more details on each entry, click the record of interest and review information in the Info Detail view.

Note: The Server Manager GUI accesses Siebel Server log files from the `log` directory of each individual Siebel Server. Siebel Server log files use the following name convention: EnterpriseServerName.SiebelServerName.log.

Monitoring Siebel Server Statistics

Monitor Siebel Server statistics using the Server Manager GUI or the Server Manager command-line interface program (srvrmgr). For background information and a list of Siebel Server statistics, see *List of Statistics and State Values*. For information about monitoring other Siebel Server run-time operations, see *Monitoring Siebel Server Status*.

To monitor Siebel Server statistics on Server Manager GUI

1. Navigate to the Administration - Server Management screen, then Servers view.
2. In the Servers list, select the Siebel Server of interest.
3. From the view tabs, click Statistics.

   Statistics for the selected Siebel Server appear in the Statistics list. For a list and description of Siebel Server statistics, see *List of Statistics and State Values*.

To monitor Siebel Server statistics on srvrmgr

- At the srvrmgr program prompt for a particular Siebel Server, enter:

```
list statistics for server siebel_server_name
```

For details on starting, running, and configuring the srvrmgr program, see *Siebel System Administration Guide* .

Monitoring Siebel Server Tasks

Monitor Siebel Server component tasks for a particular Siebel Server by using the Server Manager GUI or the Server Manager command-line interface program (srvrmgr).

For details on Siebel Server component task states, see *About Siebel Server Task States*. For information about monitoring other Siebel Server run-time operations, see *Monitoring Siebel Server Status*.

To monitor Siebel Server tasks on Server Manager GUI

1. Navigate to the Administration - Server Management screen, then Servers view.
2. In the Servers list, select the Siebel Server of interest.
3. From the view tabs, click Tasks.
4. Review the status of the tasks for the selected Siebel Server by viewing the State (Icon), State, and Status fields.

For more information about monitoring individual tasks, note the Task number and see *Monitoring Server Component Task Status*.

To monitor Siebel Server tasks on srvrmgr

- At the srvrmgr program prompt, enter:

```
list tasks for server siebel_server_name
```

For details on starting, running, and configuring the srvrmgr program, see *Siebel System Administration Guide* .

Monitoring Siebel Server User Sessions

Monitor user sessions for a particular Siebel Server by using the Server Manager GUI or the Server Manager command-line interface program (srvrmgr).

For background information about user sessions, see *About User Sessions*. For information about monitoring other Siebel Server run-time operations, see *Monitoring Siebel Server Status*.

To monitor Siebel Server user sessions on Server Manager GUI

1. Navigate to the Administration - Server Management screen, then Servers view.
2. In the Servers list, select the Siebel Server of interest.
3. From the view tabs, click Sessions.
4. Review the status of the users' sessions for the selected Siebel Server by viewing the State (Icon), Task Hung State, and State fields.

For more details on monitoring individual user sessions, note the Session ID number and see *Monitoring User Session Status*.

To monitor Siebel Server user sessions on srvrmgr

- At the srvrmgr program prompt, enter:

```
list sessions for server siebel_server_name
```

For details on starting, running, and configuring the srvrmgr program, see *Siebel System Administration Guide* .

Monitoring Siebel Server Component Status

Monitor the status of Siebel Server components by using the Server Manager GUI or Server Manager command-line interface program (srvrmgr). The following topics describe procedures that monitor the Siebel Server components:

- *Monitoring Siebel Server Component State*
- *Monitoring Siebel Server Component State Values*
- *Monitoring Siebel Server Component Statistics*

- *Monitoring Siebel Server Component Tasks*

For background information about Siebel Server components, including running and configuring procedures, see *Siebel System Administration Guide* .

Monitoring Siebel Server Component State

Monitor the status of Siebel Server components using the Server Manager GUI or the Server Manager command-line interface program (srvrmgr).

For details on Siebel Server component states, see *About Siebel Server Component States*. For information about monitoring other Siebel Server component run-time operations, see *Monitoring Siebel Server Component Status*.

To monitor the Siebel Server component state on Server Manager GUI

1. Navigate to the Administration - Server Management screen, then Components view.
2. In the Components list, select the Siebel Server component of interest.
3. Review the state of the selected Siebel Server component by viewing the State (Icon) and State fields.

The Components list view lists the Siebel Server components from all Siebel Servers operating in the Siebel Enterprise Server.

To monitor the component state on srvrmgr

- At the srvrmgr program prompt, enter:

```
list component
```

For details on starting, running, and configuring the srvrmgr program, see *Siebel System Administration Guide* .

Monitoring Siebel Server Component State Values

Monitor Siebel Server component state values using the Server Manager GUI or the Server Manager command-line interface program (srvrmgr). For background information and a list of Siebel Server state values, see *List of Statistics and State Values*. For information about monitoring other Siebel Server component run-time operations, see *Monitoring Siebel Server Component Status*.

To monitor component state values on Server Manager GUI

1. Navigate to the Administration - Server Management screen, then Components view.
2. In the Components list, select the Siebel Server component of interest.
3. From the view tabs, click State Values.

   State values for the selected Siebel Server component appear in the State Values list. For a list and description of Siebel Server state values, see *List of Statistics and State Values*.

To monitor component state values on srvrmgr

- At the srvrmgr program prompt, enter:

```
list state values for component component_alias_name
```

For details on starting, running, and configuring the srvrmgr program, see *Siebel System Administration Guide* .

Monitoring Siebel Server Component Statistics

Monitor Siebel Server component statistics using the Server Manager GUI or the Server Manager command-line interface program (srvrmgr). For background information and a list of Siebel Server component statistics, see *List of Statistics and State Values*. For information about monitoring other Siebel Server component run-time operations, see *Monitoring Siebel Server Component Status*.

To monitor component statistics on Server Manager GUI

1. Navigate to the Administration - Server Management screen, then Components view.
2. In the Components list, select the Siebel Server component of interest.
3. From the view tabs, click Statistics.

   Statistics for the selected Siebel Server component appear in the Statistics list. For a list and description of Siebel Server statistics, see *List of Statistics and State Values*.

To monitor component statistics on srvrmgr

- At the srvrmgr program prompt, enter:

```
list statistics for component component_alias_name
```

For details on starting, running, and configuring the srvrmgr program, see *Siebel System Administration Guide* .

Monitoring Siebel Server Component Tasks

Monitor tasks for a particular Siebel Server component by using the Server Manager GUI or the Server Manager command-line interface program (srvrmgr). For details on Siebel Server component task states, see *About Siebel Server Task States*. For information about monitoring other Siebel Server run-time operations, see *Monitoring Siebel Server Status*.

To monitor Siebel Server tasks on Server Manager GUI

1. Navigate to the Administration - Server Management screen, then Components view.
2. In the Components list, select the Siebel Server component of interest.
3. From the view tabs, click Tasks.
4. Review the status of tasks for the selected Siebel Server component by viewing the State (Icon), State, and Status fields.

For more details on monitoring individual tasks, note the Task number and see *Monitoring Server Component Task Status*.

To monitor component tasks on srvrmgr

- At the srvrmgr program prompt, enter:

```
list tasks for component component_alias_name
```

For details on starting, running, and configuring the srvrmgr program, see *Siebel System Administration Guide* .

Monitoring Server Component Task Status

Monitor the status of Siebel Server component tasks by using the Server Manager GUI or Server Manager command-line interface program (srvrmgr). The following topics describe procedures that monitor Siebel Server component tasks:

- *Monitoring Server Component Task State*
- *Monitoring Server Component Task Log Files*
- *Monitoring Server Component Task State Values*
- *Monitoring Server Component Task Statistics*

A task, in the context of a Siebel application, is an instantiation of a Siebel Server component. Administrators start tasks by creating jobs. Tasks are also started by the Siebel application itself. For background information about Siebel Server component tasks, including running and configuring procedures, see *Siebel System Administration Guide* .

Monitoring Server Component Task State

Monitor the state of Siebel Server component tasks using the Server Manager GUI or the Server Manager command-line interface program (srvrmgr). For details on Siebel Server component task states, see *About Siebel Server Task States*. For information about monitoring other task run-time operations, see *Monitoring Server Component Task Status*.

To monitor tasks on Server Manager GUI

1. Navigate to the Administration - Server Management screen, then Tasks view.
2. In the Tasks list, select the task of interest.
3. Review the state of the selected task by viewing the State (Icon), State, and Status fields.

The Tasks view lists tasks from all Siebel Servers operating in the Siebel Enterprise Server. To isolate tasks on a particular Siebel Server, see *Monitoring Siebel Server Tasks*. To isolate tasks for a particular Siebel Server component, see *Monitoring Siebel Server Component Tasks*.

Note: You cannot sort tasks from different Siebel Servers across the enterprise.

To monitor tasks on srvrmgr

- At the srvrmgr program prompt, enter:

```
list tasks
```

For details on starting, running, and configuring the srvrmgr program, see *Siebel System Administration Guide* .

To list tasks that have exited in error

1. Make sure that the SvrTaskPersist component (which belongs to the SystemAux component group) is enabled.
2. Run an SQL statement to query tasks that exit in error for a specific table.
   For example, you might use the following query to return tasks with errors from the S_SRM_TASK_HIST table:

   ```
   select CREATED, SRVR_PROC_ID_VAL, SRVR_LOGFILE_NAME, SRVR_STATUS
    from SIEBEL.S_SRM_TASK_HIST
    where SRVR_TASK_ID_VAL='123456789';
   ```

   All tasks that exited in error are returned by the SQL statement with the status set to ERROR.

Monitoring Server Component Task Log Files

Monitor the log files for a Siebel Server component task using the Server Manager GUI. Also review task log files by manually accessing the file or querying the file with the Log File Analyzer (LFA) utility.

For background information about:

- Event logging, see *Configuring Siebel Server and Component Logging*.
- Task log files, see *Configuring Siebel Server Component Logging*.
- LFA, see *Querying System Log Files*.

For information about monitoring other task run-time operations, see *Monitoring Server Component Task Status*.

To monitor task log files on Server Manager GUI

1. Navigate to the Administration - Server Management screen, then Tasks view.
2. In the Tasks list, select the task of interest.
3. From the view tabs, click Log.
   Each entry in the Log view list represents an event logged in the task log file.

Monitoring Server Component Task State Values

Monitor Siebel Server component task state values using the Server Manager GUI or the Server Manager command-line interface program (srvrmgr). For background information and a list of task state values, see . For information about monitoring other task run-time operations, see *Monitoring Server Component Task Status*.

To monitor task state values on Server Manager GUI

1. Navigate to the Administration - Server Management screen, then Tasks view.
2. In the Tasks list, select the task of interest.
3. From the view tabs, click State Values.
   State values for the selected task appear in the State Values list. For a list and description of task state values, see *List of Statistics and State Values*.

To monitor task state values on srvrmgr

- At the srvrmgr program prompt, enter:

```
list state values for task task_number
```

For details on starting, running, and configuring the srvrmgr program, see *Siebel System Administration Guide* .

Monitoring Server Component Task Statistics

Monitor Siebel Server component task statistics using the Server Manager GUI or the Server Manager command-line interface program (srvrmgr). For background information and a list of task statistics, see *List of Statistics and State Values*. For information about monitoring other task run-time operations, see *Monitoring Server Component Task Status*.

To monitor task statistics on Server Manager GUI

1. Navigate to the Administration - Server Management screen, then Tasks view.
2. In the Tasks list, select the task of interest.
3. From the view tabs, click Statistics.

   Statistics for the selected task appear in the Statistic list. For a list and description of task statistics, see *List of Statistics and State Values*.

To monitor task statistics on srvrmgr

- At the srvrmgr program prompt, enter:

```
list statistics for task task_number
```

For details on starting, running, and configuring the srvrmgr program, see *Siebel System Administration Guide* .

Monitoring Component Job Status

Monitor the status of Siebel Server component jobs using the Server Manager GUI. For background information about starting Siebel Server component jobs, see *Siebel System Administration Guide* . For information about component job states, see *About Component Job States*.

To monitor component job status

1. Navigate to the Administration - Server Management screen, then Jobs view.
2. In the Jobs list, select the component job of interest.
3. Review the status of the component job by viewing the Status field.

To monitor component job status requested by your User ID

1. Navigate to the Jobs screen.
2. In the My Jobs list, select the component job of interest.
3. Review the status of the component job by viewing the status field.

Monitoring User Session Status

Monitor the status of user sessions by using the Server Manager GUI or Server Manager command-line interface program (srvrmgr). The following topics describe procedures that monitor user sessions:

- *Monitoring User Session State*
- *Monitoring User Session Log Files*
- *Monitoring User Session State Values*
- *Monitoring User Session Statistics*

For background information about user sessions, see *About User Sessions*.

Monitoring User Session State

Monitor the state of Siebel Server user sessions using the Server Manager GUI or the Server Manager command-line interface program (srvrmgr). The state of the user session is that of the associated Siebel Server component task that represents the user session.

For background information about user sessions, see *About User Sessions*. For background information about Siebel Server component task states, see *About Siebel Server Task States*. For information about monitoring other Siebel Server user session run-time operations, see *Monitoring User Session Status*.

To monitor user sessions on Server Manager GUI

1. Navigate to the Administration - Server Management screen, then Sessions view.
2. In the Sessions list, select the Siebel Server user session of interest.
3. Review the state of the selected Siebel Server user session by viewing the State (Icon), Task Hung State, and State fields.

The Sessions view lists Siebel Server user sessions from all Siebel Servers operating in the Siebel Enterprise Server. To isolate sessions on a particular Siebel Server, see *Monitoring Siebel Server Tasks*.

To monitor user sessions for a Siebel Server using srvrmgr

- At the srvrmgr program prompt, enter:

```
list sessions for server siebel_server_name
```

To monitor user sessions for a Siebel Server component using srvrmgr

- At the srvrmgr program prompt, enter:

```
list sessions for comp component_alias_name
```

To monitor user sessions for a Siebel Application Object Manager using srvrmgr

- At the srvrmgr program prompt, enter:

```
list sessions for login object_manager_login
```

To list user sessions that are not responding using srvrmgr

- At the srvrmgr program prompt, enter:

```
list hung sessions for server siebel_server_name [or]comp component_alias_name
[or]login object_manager_login
```

To list active user sessions using srvrmgr

- At the srvrmgr program prompt, enter:

```
list active sessions for server siebel_server_name [or]comp component_alias_name
[or]login object_manager_login
```

For more information about starting, running, and configuring the srvrmgr program, see *Siebel System Administration Guide* .

Monitoring User Session Log Files

Monitor the log files for Siebel Server user sessions using the Server Manager GUI. User session log files are those of the associated Siebel Server component task that represents the user session. Also review Siebel Server user session log files by accessing the associated task log file or querying the associated task log file with the Log File Analyzer utility.

For background information about:

- User sessions, see *About User Sessions*.
- Siebel Server component task log files, see *Configuring Siebel Server Component Logging*.
- Log File Analyzer, see *Querying System Log Files*.
- Event logging, see *Configuring Siebel Server and Component Logging*.

For information about monitoring other Siebel Server user session run-time operations, see *Monitoring User Session Status*.

To monitor user session log files on Server Manager GUI

1. Navigate to the Administration - Server Management screen, then Sessions view.
2. In the Sessions list, select the Siebel Server user session of interest.
3. From the view tabs, click Log.

Each entry in the Log view represents an event logged in the Siebel Server component task log file, which represents the user session.

Monitoring User Session State Values

Monitor Siebel Server user session state values using the Server Manager GUI or the Server Manager command-line interface program (srvrmgr). User session state values are those of the associated Siebel Server component task that represents the user session.

For background information about user sessions, see *About User Sessions*. For background information and a list of task state values, see *List of Statistics and State Values*. For information about monitoring other Siebel Server user session run-time operations, see *Monitoring User Session Status*.

To monitor user session state values on Server Manager GUI

1. Navigate to the Administration - Server Management screen, then Sessions view.
2. In the Sessions list, select the Siebel Server user session of interest.
3. From the view tabs, click State Values.

   State values for the selected task that represent the user session appear in the State Values list. For a list and description of task state values, see *List of Statistics and State Values*.

To monitor user session state values on srvrmgr

- Use the srvrmgr command to list task state values as described in *Monitoring Server Component Task State Values*. Use the Session ID for the task number parameter in this command.

Monitoring User Session Statistics

Monitor Siebel Server user session statistics using the Server Manager GUI or the Server Manager command-line interface program (srvrmgr). User session statistics are those of the associated Siebel Server component task that represents the user session.

For background information about user sessions, see *About User Sessions*. For background information and a list of task statistics, see *List of Statistics and State Values*. For information about monitoring other Siebel Server user session run-time operations, see *Monitoring User Session Status*.

To monitor user session statistics on Server Manager GUI

1. Navigate to the Administration - Server Management screen, then Sessions view.
2. In the Sessions list, select the Siebel Server user session of interest.
3. From the view tabs, click Statistics.

   State values for the selected task that represent the user session appear in the State Values list. For a list and description of task state values, see *List of Statistics and State Values*.

To monitor user session statistics on srvrmgr

- Use the srvrmgr command to list task statistics as described in *Monitoring Server Component Task State Values*. Use the Session ID for the task number parameter in this command.

Analyzing System Data with Siebel Run-Time Data

Analyze operating system data with Siebel run-time data using the following procedures.

- *Identifying Task Log Files From the Siebel Server Log File*
- *Process of Mapping Tasks with Operating System Data*
- *Mapping User Sessions to Siebel Servers or Application Object Managers*

Identifying Task Log Files From the Siebel Server Log File

Map the Siebel Server log file to its Siebel Server components and their log files by identifying the task ID in the Siebel Server log file. Review the task log file for more information about the task performance.

> **Note:** The detail of the log file depends on logging levels set for event types for each component. For details on event types and event logging, see *Configuring Siebel Server and Component Logging*.

For information about analyzing other Siebel application diagnostic data, see *Analyzing System Data with Siebel Run-Time Data*.

To identify task IDs from Siebel Server log files

1. Access a Siebel Server log file by using the Server Manager GUI. For details on this procedure, see *Monitoring Siebel Server Log Files*.

   Also access Siebel Server log files by:

   - Using the Log File Analyzer. For details on this procedure, see *Querying System Log Files*.
   - Opening the log file itself. For details on locations and naming convention of Siebel Server log files, see *About Siebel Server Log Files*.
2. Review the Text field of each log file entry for the Siebel Server component of interest.
3. The text field of each Siebel Server component log file entry also contains the task ID number started for this component.
4. Access the Siebel Server component task list. For details on this procedure, see *Monitoring Server Component Task State*.
5. Query the list with the task ID number identified in the Siebel Server log file.
6. Review the status of the Siebel Server component task by reviewing the log file, state value, and statistics for this task. For details on these procedure, see *Monitoring Server Component Task Status*.

   > **Note:** The task ID number identified in Step 3 can also be used to find the individual task log file stored in the `log` directory. The name of the task log file contains the task ID for the component. For example, in SCCObjMgr_enu_19369.log, the task ID is 19369.

Process of Mapping Tasks with Operating System Data

Mapping tasks to operating system data allows you to view operating system CPU and memory usage for each task. Once you map a task to an operating system process ID, you can use operating system tools, such as task manager on Windows or the ps (process list) function on UNIX systems, to view other information about the process and task including CPU utilization, memory usage, and so on.

Note: Multithreaded components can have several tasks mapped to a single operating system process ID, so that the operating system tools do not necessarily break down the data by task.

Map the Siebel Server component task to the operating system data by:

1. Identifying the operating system process ID (PID) for a task. For this procedure, see *Identifying Operating System PID for a Task*.
2. Reviewing the PID in the operating system. For this procedure, see *Identifying Operating System PID for a Task*.

For information about analyzing other Siebel application diagnostic data, see *Analyzing System Data with Siebel Run-Time Data*.

Identifying Operating System PID for a Task

Identifying operating system PID numbers is a task in the *Process of Mapping Tasks with Operating System Data*. Identify operating system process ID numbers (PID) for tasks by one of the following methods:

- From the Server Manager GUI
- From the Siebel Server log file
- From the Task log file

Note: PIDs are only available in the Server Manager for running tasks.

To identify operating system PID for a task from the Server Manager GUI

1. Access the Siebel Server component task list. For details on this procedure, see *Monitoring Server Component Task State*.
2. Query the task list for a specific Siebel Server component task or task ID.
3. Note the value in the PID field for that particular task.

To identify operating system PID for a task from a Siebel Server log file

1. Access a Siebel Server log file by using the Server Manager GUI. For details on this procedure, see *Monitoring Siebel Server Log Files*.

   Also access Siebel Server log files by:

   - Using the Log File Analyzer. For details on this procedure, see *Querying System Log Files*.
   - Opening the log file itself. For details on locations and naming convention of Siebel Server log files, see *About Siebel Server Log Files*.
2. Review the Text field of each log file entry for the Siebel Server component of interest.
3. The Text field of each Siebel Server component log file entry also contains the process ID number started for this component task.

To identify operating system PID for a task from a task log file

1. Access the Siebel Server component task log file of interest. For details on locations and naming convention of Siebel Server component task log files, see *Configuring Siebel Server and Component Logging*.
2. The first entry of the task log file contains the header information. The header information contains the PID number. For a parsing of the header file and to identify the PID number, see *About Event Attributes and Log File Format*.

Reviewing the PID in the Operating System

Reviewing the process ID number in the operating system allows the identification of CPU and memory usage for individual tasks. To identify the PID number for a task, see *Identifying Operating System PID for a Task*.

Reviewing the PID numbers in the operating system is a task in the *Process of Mapping Tasks with Operating System Data*.

To review PID numbers on Microsoft Windows

1. Using the right mouse button, click a blank area on the taskbar.
2. Choose Task Manager.

   The Windows Task Manager dialog box appears.
3. Select the Processes tab and query for the task PID number.

> **Note:** If the PID column is not visible, then click View, then Select Columns.

To review PID numbers on UNIX

- Enter the command:

```
ps -ef | grep
PID
```

or:

```
ps -aux PID
```

In this command, PID is the process ID number of interest.

Mapping User Sessions to Siebel Servers or Application Object Managers

Map user sessions from the Siebel Application Interface to individual Siebel Servers or Siebel Application Object Managers by accessing the user session cookie in the Siebel Application Interface log file. For information about analyzing other Siebel application diagnostic data, see *Analyzing System Data with Siebel Run-Time Data*.

To map user session to a Siebel Server

1. Access the Siebel Application Interface log file. For details on locations and naming convention of the Siebel Application Interface log file, see *Configuring Siebel Server and Component Logging*.
2. Start the Server Manager command-line interface program (srvrmgr) at the enterprise level.

   For information about starting and running srvrmgr, see *Siebel System Administration Guide* .
3. Enter the following command:

```
list servers show SBLSRVR_NAME, SV_SRVRID
```

To map user session to a Siebel Application Object Manager task

1. Access the Siebel Application Interface log file. For details on locations and naming convention of the Siebel Application Interface log file, see *Configuring Siebel Server and Component Logging*.
2. Access the Siebel Server component task list. For details on this procedure, see *Monitoring Server Component Task State*.
3. Query the task list for the specific PID to isolate the Application Object Manager task for that user session.
4. Review data on that Application Object Manager task.

For details on these procedures, see *Monitoring Server Component Task Status*.

About Using SQL Tagging to Trace Long-Running Queries in Siebel CRM

The SQL tagging feature in Siebel CRM provides administrators with the ability to trace the origin of long-running or slow-performing SQL statements (queries) back to a specific task and user who triggered it. This topic describes the SQL tagging feature and syntax. It also provides sample code.

After SQL tagging is enabled, tagging information is added to the SELECT statements that are generated by the Siebel Application Object Manager. When a poorly performing SQL statement is suspected, the database administrator can use this tagging information at the database level to trace which component task and user initiated the SQL without the need to reboot any Siebel Server or components. Database administrators can then find the component task log file for more in-depth analysis of the performance issue. For information about enabling SQL tagging, see *Enabling and Disabling SQL Tagging*.

> **Note:** Other SQL statements generated by Siebel Application Object Manager, such as INSERT, UPDATE, and DELETE, are not tagged.

SQL Tagging Format

SQL tagging information is formatted as a comma-separated list of values using the following syntax:

```
componentname,servername,taskid,userid,flowid:sarmid,busobjname,buscompname,viewname
```

where:

- componentname is the alias of the component, for example, SCCObjMgr_enu.
- servername is the name of the Siebel Server on which the component or task is running.
- taskid is the task ID of the user who generated the query.
- userid is the login name of the user who generated the query.
- flowid is the flow ID of the component or task.
- sarmid is the SARM ID of the component or task.
- busobjname is the business object name.
- buscompname is the business component name.
- viewname is the view name (only in UI mode).

Note: Any optional elements of a tag that are irrelevant or missing for the query are replaced with an empty string.

Sample SQL-Tagged Code

The following is a sample of how a tagged SQL statement might appear in a log file when the SQL statement was generated from a Siebel Call Center (SCCObjMgr_enu) Application Object Manager component and an Oracle database. The changes made by the SQL tagging feature appear in italics.

```
SELECT
 FIRST_NAME,
 LAST_NAME,
 ...,
:1
FROM
 TBO.S_CONTACT
...
WHERE
 LAST_NAME LIKE:2
ORDER BY
...
Bind variable 1:
SCCObjMgr_enu,sdchs20i046,10485776,SADMIN,00000089489108a8:50557,Account,Account,
Account List View
Bind variable 2: Foo*
```

For information about enabling SQL tagging, see *Enabling and Disabling SQL Tagging*. For information about setting log levels for SQL tagging, see *Setting Log Levels for SQL Tagging*.

Enabling and Disabling SQL Tagging

The SQL tagging feature in Siebel CRM is a diagnostic tool that allows administrators to trace long-running or slow-performing queries back to the user or action that triggered it. For more information about SQL tagging, see *About Using SQL Tagging to Trace Long-Running Queries in Siebel CRM*.

By default, SQL tagging is disabled. Administrators can enable SQL tagging by setting the OM SQL Tagging (alias is ObjMgrSqlTag) server component event for a Siebel Application Object Manager server component, such as Call Center

Object Manager (ENU). The OM SQL Tagging event is available to all object manager-based components, such as AppObjMgr, EAIObjMgr, BusSvcMgr, and so on. You can enable and disable SQL tagging at any time if the Application Object Manager server component is running.

Enabling SQL Tagging

Use the following procedure to enable SQL tagging.

To enable SQL tagging

1. Navigate to the Administration - Server Configuration screen, then the Components view.
2. In the Components list, select the appropriate Siebel Application Object Manager for the application that you want to enable SQL tagging.

   For example, if the application you are using is Siebel Call Center, then select Call Center Object Manager (ENU).
3. Click the Events subview, and then select the OM SQL Tagging event type.
4. Set the log level to a number greater than 1 (one).

   For more information about setting the log levels, see *Setting Log Levels for SQL Tagging*.

Disabling SQL Tagging

Use the following procedure to disable SQL tagging.

To disable SQL tagging

1. Navigate to the Administration - Server Configuration screen, Components view.
2. In the Components list, select a Siebel Application Object Manager, for example, Call Center Object Manager (ENU).
3. Click the Events subview, and then select the OM SQL Tagging event type.
4. Set the log level to 1 (one).

About Setting Log Levels for SQL Tagging

The Object Manager SQL Log (alias ObjMgrSqlLog) server component event type has an SqlTag event subtype. The SqlTag subtype sets the log level for an SQL tag at a lower lever than where SQL statements are typically logged. It enables logging of the SQL tags in the log file without logging the complete SQL statements.

The OM SQL Logging (ObjMgrSqlLog) and OM SQL Tagging (ObjMgrSqlTag) events are independent of each other and can coexist. The ObjMgrSqlLog event controls the level of SQL logging detail in an object manager log file. The ObjMgrSqlTag event controls whether SQL statements are tagged or not and how much tagging information is generated, irrespective of whether or not the SQL statements are logged in the log file. For example, if the ObjMgrSqlLog event log level is set to 1, then neither the SQL statements nor the SQL tags are logged in the log files, even if the ObjMgrSqlTag subevent is active. Whereas, if the ObjMgrSqlLog event log level is set to 4, then full details about the SQL information is generated in the log files. However, if the ObjMgrSqlTag subevent is not active, then the SQL is not tagged, nor are the SQL tags logged in the log files.

If SQL tagging is active and the SQL Logging event is set at the SqlTag level, then only the SQL tags are logged in the log files as shown in the following example:

```
Begin: Execute SqlObj 'Account' at 11160f10 with
SqlTag=SCCObjMgr_enu,sdchs20i046,10485776,,SADMIN,00000089489108a8:50557,
Account,Account,Account List V
```

This configuration is useful in diagnosing long-running SQLs without generating too much SQL logging in the log files.

> **Note:** The END statement is logged so that you can run a script to identify log files where the BEGIN SQL tag statement is present, but there is no END statement. This syntax helps determine which log files might contain long-running SQL statements.

For information about setting log levels for SQL tagging, see *Setting Log Levels for SQL Tagging*.

Setting Log Levels for SQL Tagging

The OM SQL Tagging (ObjMgrSqlTag) server component event controls whether SQL tagging is enabled and how much tagging information is generated. Event log levels (event subtypes) control the level of detail that is added to each tag. The higher the log level, the higher the level of detail. For more information about setting SQL tagging log levels, see *About Setting Log Levels for SQL Tagging*.

To set the log level for SQL tagging

1. Navigate to the Administration - Server Configuration screen, then the Components view.
2. In the Components list, select the appropriate Siebel Application Object Manager for the application that you want to enable SQL tagging.

   For example, if the application you are using is Siebel Call Center, then select Call Center Object Manager (ENU).
3. Click the Events subview, and then select the OM SQL Tagging event type.
4. Change the log level to a number using the following guidelines.

Log Level	Description
0	SQL tagging is disabled.
1	SQL tagging is disabled.
2	SQL tagging includes only the component name, server name, and task ID. **Note:** Use this log level when you do not want user IDs exposed in an SQL Tagging event log for an Application Object Manager.
3	SQL tagging includes the component name, server name, task ID, user ID, and flow ID with the SARM ID.

Log Level	Description
4	SQL tagging includes the component name, server name, task ID, user ID, and flow ID with the SARM ID, business object, business component, and view name.

About Siebel Process Failure Diagnostics

Siebel CRM provides process failure diagnostics that help you to allocate system resources and maintain an efficient and reliable environment. Administrators can monitor KPIs (key performance indicators) of the various levels of the Siebel environment to do the following:

- Identify, isolate, and remedy adverse system and application conditions.
- Adjust the various settings and parameters of the Siebel environment to avert adverse conditions.
- Optimize available resources for improved performance.

When a Siebel process fails, the administrator can do the following:

- View all detectable failed processes for a given Siebel Enterprise by Siebel Server.
- Identify the probable cause and point of origin of the failure, including (but not limited to) user, process, thread, task, application, view, and activity immediately prior to the failure.
- Review content of the main process failure file (crash.txt) associated with a given failure including the call stack, the register contents, and the memory information.
- Review content of other failure information contained in the Siebel FDR (Flight Data Recorder), Siebel ARM (Application Response Measurement), and component log files.
- Determine other Siebel users directly affected by the failure.

Related Topics

How Siebel Process Failure Diagnostics Work

Scenario for Working with Siebel Process Failure Diagnostics

Investigating Failed Siebel Server Processes

Example of Investigating a Failed Siebel Server Process

How Siebel Process Failure Diagnostics Work

Siebel process failure diagnostics collects data for use by administrators to diagnose and troubleshoot a variety of failed Siebel Server processes.

The SvrTaskPersist component of the SystemAux component group handles the diagnostic data collection. This component uses the SIEBEL_DIAG_STORE environment variable as a location store to retrieve the diagnostic data (FDR file, crash.txt file, component log file, failure summary, and so on).

The administrator can use the siebprocdiag command-line utility to output the process failure data to any path you specify for future retrieval. The utility scans the `SIEBEL_ROOT\siebsrvr\bin` directory and collects the various files for each process failure and copies them to the specified directory. For example, if you execute the following command for Windows, then the process failure files are copied to `d:\temp`:

```
siebprocdiag d:\temp
```

For information about configuring system environment variables and using server management utilities, see *Siebel System Administration Guide* .

Related Topics

About Siebel Process Failure Diagnostics

Scenario for Working with Siebel Process Failure Diagnostics

Investigating Failed Siebel Server Processes

Example of Investigating a Failed Siebel Server Process

Scenario for Working with Siebel Process Failure Diagnostics

This topic provides a scenario for working with Siebel Process Failure Diagnostics. You might use this feature differently, depending on your business needs.

An administrator is informed by a user that the Siebel application is unresponsive. The administrator navigates to the Process Failure Diagnostics view to investigate whether the cause might be a failed process. After identifying a failed process that is associated with that user, the administrator can view the details of the failure including:

- Siebel Server and server component names
- Time of failure
- Process, thread, and task IDs
- Number of affected tasks
- Last set of meaningful business processes that occurred at the time of the failure
- Whether a new process was created, and if so, the list of current users impacted by the failure
- A list of users whose sessions were lost following the failure
- Content of the actual crash.txt file that was logged, which provides the call stack, register contents, and memory information

The administrator then determines the cause and effect to address the issue or investigates further to resolve. When applicable, the administrator forwards the details to an internal technical support team to assist them in their troubleshooting activities. The administrator might also want to query the failed process data in the Siebel database to generate histograms or reports to initiate preventative measures.

Related Topics

About Siebel Process Failure Diagnostics

How Siebel Process Failure Diagnostics Work

Investigating Failed Siebel Server Processes

Example of Investigating a Failed Siebel Server Process

Investigating Failed Siebel Server Processes

The process failure feature in Siebel CRM provides administrators with the ability to analyze and diagnose various system failures. Administrators can identify the following:

- Siebel Server and component process or processes that have failed
- Users who have lost sessions as a result of the process failure
- User actions that might have resulted in the failure
- Dump files associated with the failed process

To investigate a failed Siebel Server process

1. Navigate to the Administration - Server Management screen, Diagnostics, and then the Process Failure Diagnostics view.
2. In the Failed Processes list, select a record to view the Siebel Server, server component, failure time, and other details about the process that failed.
3. In Affected Users on Failed Process, view the users who have lost the sessions as result of the process failure.
4. In Failed User Task, view the user actions that might have resulted in the failure.
5. In Failed Process Call Stack, view the content of the crash.txt file, which is the call stack information.

If you need further assistance with troubleshooting, then create a service request (SR) on My Oracle Support. Alternatively, you can phone Oracle Global Customer Support directly to create a service request or get a status update on your current SR. Support phone numbers are listed on My Oracle Support.

Related Topics

About Siebel Process Failure Diagnostics

How Siebel Process Failure Diagnostics Work

Scenario for Working with Siebel Process Failure Diagnostics

Example of Investigating a Failed Siebel Server Process

Example of Investigating a Failed Siebel Server Process

This topic gives one example of how you might investigate a failed Siebel Server process. You might use this feature differently, depending on your business needs. When a Siebel Server process failure is suspected, administrators can diagnose the nature of the failure in detail.

To investigate a failed Siebel Server process

1. Navigate to the Administration - Server Management screen, then the Process Failure Diagnostics view.
2. In the Failed Process list, select the failed process for which you want to learn more, then review the details about that failure. Details include: the Siebel Server and server component on which the process failed; the time of failure; process, thread, and task IDs; the location of related failure dump files, and so on.
3. Identify all users associated with this task, and notify these users as appropriate, that a key process related to their current activity has failed. Both the administrator and users can then take appropriate action, such as restarting any operations that have halted because of the failed process.
4. Review the activity and events (user-initiated and otherwise) that occurred immediately prior to the process failure.

   Administrators can use this information to detect patterns in Siebel configurations, usage, and interdependencies of Siebel entities (components, attribute and parameter settings, hardware, and so on) that might lead to process failures.
5. Query the failed process data in the Siebel database to generate histograms and reports. If the data indicates that failures are related to resources or throughput, then administrators can use that data for potential preventative measures.
6. Forward details from failure-related files and detectable patterns to technical support to assist in their troubleshooting of code and configuration issues.

Related Topics

About Siebel Process Failure Diagnostics

How Siebel Process Failure Diagnostics Work

Scenario for Working with Siebel Process Failure Diagnostics

Investigating Failed Siebel Server Processes

4 Configuring Siebel Server and Component Logging

Configuring Siebel Server and Component Logging

This chapter provides descriptions and examples of configuring Siebel Server and component logging using Siebel events. It includes the following topics:

About Configuring Siebel Server and Component Logging

Configuring Siebel Server and component logging captures the internal activity and behavior of Siebel CRM during operation. Siebel Server and component logging use event logging to collect data and write the information to a text log file. You can configure event logging to use system alerts, or you can use event logging with third-party system management applications to notify administrators of any significant or adverse conditions. For information about configuring server components to use system alerts, see *Siebel System Administration Guide* . You can monitor and manage most Siebel CRM products and functional areas with event logging.

The information collected by event logging can range from error messages to detailed diagnostic logs. Some of the application conditions and operations that result in data written to the log file include:

- Catastrophic or error conditions
- Change of status of a Siebel Server or server component
- Start or finish of a Siebel process or workflow
- Specific point in a Siebel process or workflow
- When measurable threshold values are reached or exceeded
- When operational conditions are met

About Events and Event Logging

The elements of event logging are defined in the following bullets:

- **Event.** An event is created each time you execute a program code (such as running a task).
- **Event Type.** Event types are categories of events.
    - For information about event types pertinent to a specific part of Siebel CRM, see product-specific documentation or details available on Siebel Bookshelf.

  - For generic event types used in server component and Siebel Application Object Manager diagnostics, see *Common Event Types for Component Diagnostics* and *Common Event Types for Siebel Application Object Manager Diagnostics*.

- **Event Subtype.** Event subtypes are code references that define the event.
- **Log Level.** The log level determines the amount of information that is written to the log file. Log levels are set for event types. The following table lists the log levels of event types.
- **Severity.** A severity level is associated with each event subtype. The severity level and log level share the same scale and are compared when writing events to the log file. The following table lists the severity of event subtypes.

Log and Severity Level	Description
0	Fatal
1	Errors
2	Warnings
3	Informational
4	Details
5	Diagnostic

When an event occurs, the severity level of the event (as defined by the event subtype) is compared with the log level of the event type. If the numerical value of the event severity level is equal to or lower than the numerical value of the event type log level, then the event is written to the log file. If the numerical value of the event severity level is higher than the numerical value of the event type log level, then the event is ignored.

Note: Event subtypes with a lower numeric value have a higher severity. For example, a value of 0 indicates that the event subtype is more severe than one with a value of 5. If you set the event log level to a low number, such as 1, then only the most severe events are logged. If you set the event log level to a higher number, such as 5, then more information is captured, including less severe event subtypes.

For example, the Siebel Server components in the Enterprise Application Integration component group (alias EAI) have an event type called EAI Siebel Wizard. Several event subtypes belong to the EAI Siebel Wizard event type, including:

- EAI Siebel Wizard Invalid Business Component with a severity level of 2
- EAI Siebel Wizard Invalid MVG with a severity level of 2
- EAI Siebel Wizard MVG with a severity level of 3

While the EAI component group is running, the process encounters a multi-value group (MVG). This encounter creates an event of the EAI Siebel Wizard MVG subtype. If the MVG is invalid, then a second event of the EAI Siebel Wizard Invalid MVG subtype is created. If the log level of the EAI Siebel Wizard event type is set to 1, then both events are ignored. If the log level is set to 3, then both events are written to the log file.

Events are logged at the Siebel Server level and the component level. For details on Siebel Server events, see *Configuring Siebel Server Logging*. For information about component events, see *Configuring Siebel Server Component Logging*.

About Event Attributes and Log File Format

Each event within the log file contains information about the associated application condition, including:

- Event Identifier
  - Type (category)
  - Subtype
- Timestamp
- Severity Level
- Details (metrics) about the event

For examples of individual events and their attribute values, see *Examples of Siebel Server Log Files* and *Examples of Component Log Files*. For an example of a group of events collected within a log file, see *Example of a Detailed Component Log File*.

Events are written to and collected in a log file in the order of their occurrence. Each log file contains a header that provides information on the individual log file. The following is an example of a log file header:

```
ï»¿2021 2017-05-07 21:02:06 0000-00-00 00:00:00 -0800 00000000 001 003f 0001 09
SiebSrvr 2049 1364 1548 C:\siebel\ses\siebsrvr\log\siebel17.server1.log 17.0
[xxxxx] ENU
```

The following table provides descriptions of the example log file header details.

Log File Header Detail	Description
`ï»¿`	Byte Order Marker (BOM). The BOM is a Unicode format instruction. If the log file header opens with characters like those shown here, then it indicates that the text editor used to view the log file cannot interpret the Unicode instruction.
`2017-05-07 21:02:06`	Timestamp of log file creation.
`-0800`	Offset of the local time from the GMT in the format ±HHMM.
`SiebSrvr`	The Siebel Server or component alias to which this log file refers.
`2049`	Task ID.
`1364`	OS Process ID (PID).
`1548`	Thread ID.
`C:\siebel\ses\siebsrv`	Log filename.

Log File Header Detail	Description
`r\log\siebel17.server 1.log`	
`17.0`	Version number.
`[xxxxx]`	Build number.
`ENU`	Language code.

About Siebel Server Log Files

Siebel Server log files record data for each individual Siebel Server deployed as part of a Siebel Enterprise Server. The Siebel application stores Siebel Server log files in the `log` directory for each individual Siebel Server as shown in the following table.

Operating System	Log Directory
Windows	`SIEBSRVR_ROOT\log`
UNIX	`SIEBSRVR_ROOT/enterprises/EnterpriseServerName/SiebelServerName/log`

Server log files use the following name convention: EnterpriseServerName.SiebelServerName.log.

Information contained in the Siebel Server log file can be used to determine where to search and investigate component log files for more information. The task ID, which makes up a part of the component log filename, is referenced in messages written to the Siebel Server log file. Locate the appropriate component task ID in the Siebel Server log file and open the task-specific component log that has the task ID in the log filename. For an example of this relationship, see *Example of Component Startup Log File*.

For more information and examples of Siebel Server log files, see *Viewing Siebel Server Log Files* and *Examples of Siebel Server Log Files*.

About Component Log Files

Siebel Server component log files record data for each individual component and task functioning on a particular Siebel Server. These component log files are stored in the Siebel Server `log` directory on the Siebel Server in which the components are active as shown in the following table. Using event logging with individual components allows you to isolate portions of a Siebel application.

Operating System	Log Directory
Windows	`SIEBSRVR_ROOT\log`

Operating System	Log Directory
UNIX	`SIEBSRVR_ROOT/enterprises/EnterpriseServerName/SiebelServerName/log`

Component log files use the following naming convention:

ComponentAliasName_SISProcID_TaskID.log

where:

- ComponentAliasName is the name of the component running the task.
- SISProcID is an internal four-character, zero-padded process ID that is rotating and incremented as component processes are spawned. The minimum numeric value allowed for SISProcID is 1. The maximum value allowed is 2047.
- TaskID is a 32-bit internal zero-padded task ID number. Internally, the task ID contains a SISProcID as well as a counter maintained in each component process.

There is one process ID counter for all processes, not for each component. Therefore, you can sort the log files of a particular component by the specific component process.

Individual component task log files can also be consolidated into a single log file by setting the Use Shared Log File (alias LogUseSharedFile) component parameter. For more information about this parameter and on administering the Siebel Server and server component parameters, see *Siebel System Administration Guide* . For more information about and examples of component log files, see *Viewing Component Log Files* and *Examples of Component Log Files*.

Configuring Siebel Server Logging

Siebel Server logging use event types that relate to Siebel Servers. For example, the Server State event type is a Siebel Server-level event that logs changes to the state of the Siebel Server. This topic describes how to configure and view Siebel Server event types. For details, see:

- *Setting Log Levels for Siebel Server Event Types*
- *Viewing Siebel Server Log Files*
- *Examples of Siebel Server Log Files*

Setting Log Levels for Siebel Server Event Types

This topic describes setting log levels for Siebel Server event types using the Server Manager GUI or Server Manager command-line interface program (srvrmgr). For background information about event logging and event types, see *About Configuring Siebel Server and Component Logging*. To see the resultant Siebel Server log files, see *Viewing Siebel Server Log Files*. For examples of Siebel Server log files, see *Examples of Siebel Server Log Files*.

Note: The log level setting takes place immediately and affects all components for that Siebel Server.

To set log levels for a Siebel Server event type on Server Manager GUI

1. Navigate to the Administration - Server Configuration, then Servers view.

2. In the Siebel Servers list, select the Siebel Server of interest.
3. From the view tabs, click Events.
4. In the Event Type list, select the Siebel Server Event Type of interest.

   For information about event types pertinent to a specific part of Siebel CRM, see product-specific documentation or details available on Siebel Bookshelf.
5. In the Log Level field, choose the log level that you want to set for this event type.

   For a list of log levels, see the table in *About Events and Event Logging*.
6. Click the menu button and then Save Record.

To set log levels for a Siebel Server event type on srvrmgr

- Enter:

```
change evtloglvl event_alias_name=level for server siebel_server_name
```

To list Siebel Server event types on srvrmgr

- Enter:

```
list evtloglvl for server siebel_server_name
```

For details on starting, running, and configuring the srvrmgr program, see *Siebel System Administration Guide* .

Viewing Siebel Server Log Files

Siebel Server-level events are written to the Siebel Server log file. The `log` directory location on Windows is `SIEBSRVR_ROOT\log`. The `log` directory location on UNIX is `SIEBSRVR_ROOT/enterprises/EnterpriseServerName/SiebelServerName/log`.

For background information about event logging and event types, see *About Configuring Siebel Server and Component Logging*. For more information and file naming conventions, see *About Siebel Server Log Files*. For examples of Siebel Server log files, see *Examples of Siebel Server Log Files*.

You can also view Siebel Server event logs from the Server Manager GUI. For information about this task, see *Monitoring Siebel Server Log Files*.

To assist in analyzing Siebel Server event log files, use the Log File Analyzer (LFA) utility to query and isolate log files of interest. For information about this feature, see *Querying System Log Files*.

Examples of Siebel Server Log Files

This topic provides examples of Siebel Server event log files. The event log format and information are detailed and described with the examples.

Example of Siebel Server Startup Log File

The following log file samples display what's written to the server log file during a regular startup of a Siebel Server. In this example, events are created that are defined by the event subtypes LstnObjCreate, ProcessCreate, and Startup, all which have a severity of 1. For a detailed description of the sample output, see the following tables. These events belong to the event type Server Logging (alias ServerLog). If this event type is set to a log level between 1 and 5, then the following information is a sample of what's recorded in the log file.

ObjMgrGenericInfo Event Subtype

The following table describes the output for an ObjMgrGenericInfo event subtype for the following entry. The ObjMgrGenericInfo is a subtype of the ObjMgrGenericLog Event Type. It is designed to display information that isn't an error but that's important information about your currently running Object Manager's configuration:

```
ObjMgrGenericLog ObjMgrGenericInfo 1 00000008658301e2:0 2023-12-21 06:35:05 Workspace Name = MAIN
ObjMgrGenericLog ObjMgrGenericInfo 1 00000008658301e2:0 2023-12-21 06:35:05 Workspace Version = 0
ObjMgrGenericLog ObjMgrGenericInfo 1 00000008658301e2:0 2023-12-21 06:35:05 Repository Version = 24.2
ObjMgrGenericLog ObjMgrGenericInfo 1 00000008658301e2:0 2023-12-21 06:35:05 Application Version = v24.2
```

Log Detail	Description
ObjMgrGenericLog	Event Type alias
ObjMgrGenericInfo	Event Subtype
1	Event Severity
0	SARM ID
2023-12-21 06:35:05	Date and time of log
Examples: Workspace Name = MAIN Workspace Version = 0 Repository Version = 24.2 Application Version = v24.2	Log message

LstnObjCreate Event Subtype

The following table describes the output for a LstnObjCreate event subtype for the following entry:

```
ServerLog LstnObjCreate 1 0 2017-05-13 11:35:10 Created port 49173 for Server
Request Processor
```

Log Detail	Description
ServerLog	Event Type alias

Log Detail	Description
LstnObjCreate	Event Subtype
1	Event Severity
0	SARM ID
2017-05-13 11:35:10	Date and time of log
Created port 49173 for Server Request Processor	Log message

Startup Event Subtype

The following table describes the output of a Startup event subtype for the following entry:

```
ServerLog Startup 1 0 2017-05-13 11:35:10 Siebel Application Server is ready and
awaiting requests
```

Log Detail	Description
ServerLog	Event Type alias
Startup	Event Subtype
1	Event Severity
0	SARM ID
2017-05-13 11:35:10	Date and time of log
Siebel Application Server is ready and awaiting requests	Log message

ProcessCreate Event Subtype

The following table describes the output of a ProcessCreate event subtype for the following entry:

```
ServerLog ProcessCreate 1 0 2017-05-13 11:35:10 Created multithreaded server process
(OS pid = 2756) for File System Manager with task id 4114
```

Log Detail	Description
ServerLog	Event Type alias

Log Detail	Description
ProcessCreate	Event Subtype
1	Event Severity
0	SARM ID
2017-05-13 11:35:10	Date and time of log
Created multithreaded server process	Log message
(OS pid = 2756)	Operating System Process ID number
for File System Manager	Siebel Server Component
with task id 4114	Task ID number referencing the Siebel Server task

Configuring Siebel Server Component Logging

Component logging uses event types that relate to a specific Siebel Server component. For example, the SQL Tracing event type is a component-level event that traces SQL statements for a particular server component. This topic describes how to configure and view server component event types. For details, see the following topics:

- *Setting Log Levels for Component Event Types*
- *Viewing Component Log Files*
- *Examples of Component Log Files*
- *Common Event Types for Component Diagnostics*
- *Common Event Types for Siebel Application Object Manager Diagnostics*

Setting Log Levels for Component Event Types

This topic describes setting log levels for server component event types using the Server Manager GUI or Server Manager command-line interface program (srvrmgr). For background information about event logging and event types, see *About Configuring Siebel Server and Component Logging*. To see the resultant Siebel Server component log files, see *Viewing Component Log Files*. For examples of Siebel Server component log files, see *Examples of Component Log Files*.

Note: The log level setting takes place immediately.

Setting Log Levels for Siebel Server Component Event Types Using the Server Manager GUI

Use the following procedure to set log levels for Siebel Server component event types using the Server Manager GUI.

To set log levels for a Siebel Server component event type using the Server Manager GUI

1. Navigate to the Administration - Server Configuration screen, then the Servers view.
2. In the Siebel Servers list, select the Siebel Server of interest.
3. Click the Components view tab.
4. In the Components list, select the Siebel Server component of interest.

   For example, you might select Call Center Object Manager (ENU).
5. Click the Events subview tab.
6. Select the Siebel Server component event type of interest.
   - For information about event types pertinent to a specific part of Siebel CRM, see product-specific documentation or details available on Siebel Bookshelf.
   - For generic event types used in server component and Siebel Application Object Manager diagnostics, see *Common Event Types for Component Diagnostics* and *Common Event Types for Siebel Application Object Manager Diagnostics*.
7. In the Log Level field, type in the log level you want to set for this event type, and then step off the record to save it.

   For a list of log levels and descriptions, see the table in *About Events and Event Logging*.

Setting Log Levels for Siebel Server Component Event Types Using srvrmgr

Use the following procedures to set log levels for Siebel Server component event types using the Server Manager command-line interface program (srvrmgr).

To configure a component event type using srvrmgr

- Enter:

```
change evtloglvl event_alias_name=level for component component_alias_name
```

To configure a server-specific component event type using srvrmgr

- Enter:

```
change evtloglvl event_alias_name=level for server siebel_server_name component
component_alias_name
```

To list component event types using srvrmgr

- Enter:

```
list evtloglvl for component component_alias_name
```

To set server component event log levels for all users using srvrmgr

1. Make sure the value for the List of users parameter (alias UserList) is empty.

   When this parameter lists one or more particular users, then extended logs are created only for those user IDs in the list.

   For information about setting log levels for Siebel Server component parameters, see *Setting Log Levels for Component Event Types*.
2. Enter:

   ```
   change evtloglvl event_alias_name=level for component component_alias_name
   ```

   Detailed log events for all users are written to the log file.

To set server component event log levels for a specific user using srvrmgr

1. Make sure the value for the List of users parameter (alias UserList) is set to SADMIN.

   For information about setting log levels for Siebel Server component parameters, see *Setting Log Levels for Component Event Types*.
2. Enter:

   ```
   change evtloglvl event_alias_name=level for component component_alias_name
   ```

   Detailed log events for only SADMIN users are written to the log file.

   Note: The log level for other users remains the same.

For details on starting, running, and configuring the srvrmgr program, see *Siebel System Administration Guide* .

Viewing Component Log Files

Component-level events are written to log files for each task based on the component. The `log` directory location on Windows is `SIEBSRVR_ROOT\log`. The `log` directory location on UNIX is `SIEBSRVR_ROOT/enterprises/EnterpriseServerName/SiebelServerName/log`. Portions of component task log files can be viewed from the Server Manager GUI. For more information, see *Monitoring Server Component Task Log Files*. Individual component task log files can also be consolidated into a single log file. For more information and file naming conventions, see *About Component Log Files*.

To assist in analyzing Siebel Server component event log files, use the Log File Analyzer (LFA) utility to query and isolate log files of interest. For information about this feature, see *Querying System Log Files*.

Examples of Component Log Files

This topic provides excerpts and examples of component event log files. The event log format and information are described with each of the examples.

Example of Component Startup Log File

The following log file sample displays what is written to the individual Siebel Server component log files during a regular startup of components running on a Siebel Server. In the following example, an event is created for the File System Manager component that is defined by the event subtype LstnObjInherit. For a detailed description of this sample output, see the following table. This event has a severity of 3 and events of this subtype belong to the event type ServerLog. If this event type is set to a log level between 1 and 5, then the following information is recorded in the log file.

```
ServerLog LstnObjInherit 3 0 2017-05-13 11:35:10 Inherited listening object for port
49172
```

Log Detail	Description
ServerLog	Event Type alias
LstnObjInherit	Event Subtype
3	Event Severity
0	SARM ID
2017-05-13 11:35:10	Date and time of log
Inherited listening object for port 49172	Log message

This sample log file extract is from the component log file named FSMSrvr_4114.log and is located in the `log` directory of the Siebel Server. The task ID, 4114, which defines this log file title, corresponds to the log message in the appropriate Siebel Server log file. For this message, see the table in *Example of Siebel Server Startup Log File*.

Example of Server Request Broker Log File

The following examples display log file entries in a sample Server Request Broker log file. The name of this log file is SRBroker_TaskID.log and is found in the Siebel Server `log` directory. The first sample captures an event defined by the event subtype GenericInfo, which belongs to the component event type General Events (alias GenericLog). For a detailed description of this sample output, see the following table. This event has a severity of 3 and is recorded to the log file if the General Event log level is set between 3 and 5.

```
GenericLog GenericInfo 3 0 2017-05-13 14:07:31 Set environment variable
DB2CODEPAGE=1252
```

Log Detail	Description
GenericLog	Event Type alias
GenericInfo	Event Subtype
3	Event Severity

Log Detail	Description
0	SARM ID
2017-05-13 14:07:31	Date and time of log
Set environment variable DB2CODEPAGE=1252	Log message

The next two samples belong to the component event type SQL Parse and Execute. Events were recorded of the event subtype Statement and Prepare + Execute. For detailed descriptions of the sample output, see the following tables. Both of these event subtypes have a severity of 4 and are recorded to the log file if the SQL Parse and Execute event type is set to either 4 or 5.

Statement Event Subtype

The following table describes the output for a Statement event subtype for the following entry:

```
SQLParseAndExecute Statement 4 0 2017-05-13 14:07:38 select ROW_ID, NEXT_SESSION,
MODIFICATION_NUM from dbo.S_SSA_ID
```

Log Detail	Description
SQLParseAndExecute	Event Type alias
Statement	Event Subtype
4	Event Severity
0	SARM ID
2017-05-13 14:07:38	Date and time of log
select ROW_ID, NEXT_SESSION, MODIFICATION_NUM from dbo.S_SSA_ID	SQL statement

Prepare + Execute Event Subtype

The following describes the output for a Prepare + Execute event subtype for the following entry:

```
SQLParseAndExecute Prepare + Execute 4 0 2017-05-13 14:07:38 Time: 0s, Rows: 0, Avg.
Time: 0s
```

Log Detail	Description
SQLParseAndExecute	Event Type alias

Log Detail	Description
Prepare + Execute	Event Subtype
4	Event Severity
0	SARM ID
2017-05-13 14:07:38	Date and time of log
Time: 0s, Rows: 0, Avg. Time: 0s	SQL Execution statistics

Example of a Log File for a Server Request Processor

The following code displays a log file entry in a sample server request processor log file, which can help you troubleshoot why a component request might not run. The name of this log file is SRProc_TaskID.log and is found in the `SIEBELSRVR_ROOT\log` directory. This code captures an event defined by the event SRMRouting subtype, which belongs to the SRMRouting component event type. The following table provides a detailed description of the sample output. This event has a log level of 4.

```
2021 2017-05-11 11:54:24 2017-05-11 12:50:55 +0530 000002d4 001 003f 0001 09
TestMTSBound 4194307 332 2976 m:\siebel\log\TestMTSBound_0004_4194307.log 17.0
[xxxxx] ENU
```

Log Header Detail	Description
2021	Indicates the values of the LogEol and LogXlateMsgs parameters, as well as the file version and file completion indicators. The first number represents the LogEol parameter. This parameter can take values CRLF, LF, CR, or a custom value where: • CRLF represents \r\n and is shown as 2 in the log file. • LF represents \n and is shown as 1 in the log file. • CR represents \r and is shown as 0 in the log file. • A custom value is shown as 3 in the log file. The second number represents the file completion indicator as zero and does not change. The third number represents the file version indicator as 2 and does not change. The fourth number represents the value of the LogXlateMsgs (translate log file) parameter in the log file. The LogXlateMsgs parameter can take the values true or false where: • *True* indicates that log files are translated. It is shown as *1* in the log file. • *False* indicates log files are translated. It is shown as *2* in the log file. In this example, 2021 indicates the following: • LogEol = 2 (new line character is \r\n) • File completion error is 0 • File version is 2

Log Header Detail	Description
	• LogXlateMsgs = 1 (translate)
2017-05-11 11:54:24	Log file creation timestamp.
2017-05-11 12:50:55	Log file completion timestamp.
+0530	Indicates the time difference of the Siebel Server time zone from GMT in the format ±HHMM, where *HH* represents hours, and *MM* represents minutes.
000002d4	The number of lines in the log file in hexadecimal format.
001	The segment number in decimal format.
003f	The LogEntryFlgs parameter value is applicable only if the LogEntryFmt parameter is set to *delimited*. If the LogEntryFmt parameter is set to *fixed*, then all the fields are logged. When the LogEntryFmt is set to delimited, the LogEntryFlgs parameter takes values in decimal format and then internally converts the values to binary format. For example, if the LogEntryFlgs parameter is set to the value of 63, then the value is converted to its binary equivalent of 00111111, and is then processed further. The bits are numbered from right to left, starting from 0 to 7 (the seventh bit is 0, and the zero-position bit is 1). Given the following log file entry: `SisnTcpIp SisnSockDetail 4 00000cd049bd8290:0 42017-05-16 10:57:02 35928:LOCALTRANS-server] accept() timeout during get conn request` If the bit in the: • Zero position is set, then this least significant bit (LSB) logs the main event type, which in the log file entry is *SisnTcpIp*. • First position is set, then the subevent type is logged, which in the log file entry is *SisnSockDetail*. • Second position is set, then the event severity is logged, which in the log file entry is *4*. • Third position is set, then the timestamp is logged, which in the log file entry is *42017-05-16 10:57:02*. • Fourth position is set, then the details are logged, which in the log file entry is *35928:LOCALTRANS-server] accept() timeout during get conn request*. • Fifth position is set, then the flow ID and the SARM ID are logged, which in the log file entry is *00000cd049bd8290*.
0001	The number of characters in the LogFieldDelim parameter. The default value is `\t`.
09	Represents the characters in the LogFieldDelim parameter. The default value is `\t`. For example, 09 represents the character `\t`, and 09 is the hexadecimal representation of the ASCII value of the character `\t`.
TestMTSBound	The name of the component.
4194307	The task ID.

Log Header Detail	Description
332	The process ID of the component process.
2976	The thread ID.
m:\siebel\log\TestMTSBound_0004_4194307.log	The log file name.
17.0 [*xxxxx*] ENU	The product version, including the language code.

Example of Component Error Log File

This example displays an error entry from a sample Assignment Manager component log file. The log file is located in the `SIEBSRVR_ROOT\log` directory and is named AsgnSrvr_TaskID.log. The log message details an event defined by the event subtype GenericError, which belongs to the component event type General Events (alias GenericLog). For a detailed description of the sample output, see the following table. An error event has a severity of 1 and is recorded to the log file if the General Event log level is set between 1 and 5.

```
GenericLog GenericError 1 0 2017-05-03 01:02:12 [MERANT][ODBC Oracle 8
driver][Oracle 8]ORA-12541: TNS:no listener
```

Log Detail	Description
GenericLog	Event Type alias
GenericError	Event Subtype
1	Event Severity
0	SARM ID
2017-05-03 01:02:12	Date and time of log
MERANT][ODBC Oracle 8 driver][Oracle 8]ORA-12541: TNS:no listener	Error message

Example of a Detailed Component Log File

The previous log file examples are sample extracts from various component log files. As a final example, the following collection of log file messages display the output recorded to a log file after a successful task run by the Document Server component. This log file information is recorded when the appropriate event type log levels are set.

```
2021 2017-05-16 23:28:38 0000-00-00 00:00:00 -0600 00000000 001 003f 0001 09 SiebSrvr 0
5956 3856 D:\siebel\ses\siebsrvr\log\siebel.sdc78275svqe.log 17.0 [xxxxx] ENU

ServerLog ServerStartup 1 0000622e49ba14c8:0 2017-05-16 23:28:38 Siebel Enterprise
Applications Server is starting up
```

```
ServerLog LstnObjCreate 1 000062d549ba14c8:0 2017-05-16 23:28:38 Created port 49156 for
Workflow Process Batch Manager

ServerLog LstnObjCreate 1 000062d549ba14c8:0 2017-05-16 23:28:38 Created port 49157 for
Workflow Recovery Manager

ServerLog LstnObjCreate 1 000062d649ba14c8:0 2017-05-16 23:28:38 Created port 49158 for
Workflow Process Manager

ServerLog LstnObjCreate 1 000062d649ba14c8:0 2017-05-16 23:28:38 Created port 49159 for File
System Manager

ServerLog LstnObjCreate 1 000062d649ba14c8:0 2017-05-16 23:28:38 Created port 49160 for
Server Request Processor

ServerLog LstnObjCreate 1 000062d649ba14c8:0 2017-05-16 23:28:38 Created port 49161 for
Siebel Administrator Notification Component


...


ServerLog ProcessExit 1 0000651d49ba14c8:0 2017-05-16 23:30:03 SmartAnswer 6612 TERMINATED
Process 6612 was terminated

ServerLog ComponentUpdate 2 000013f949bf1744:0 2017-05-16 23:30:07 CommOutboundMgr
 INITIALIZED Component has initialized.

ServerLog ProcessCreate 1 000013f949bf1744:0 2017-05-16 23:30:15 Created server process (OS
pid = 2660 ) for ServerMgr

ServerLog ProcessCreate 1 000013f949bf1744:0 2017-05-17 00:45:51 Created server process (OS
pid = 7624 ) for ServerMgr

ServerLog ProcessCreate 1 000013f949bf1744:0 2017-05-17 03:43:39 Created server process (OS
pid = 3236 ) for ServerMgr

ServerLog ProcessExit 1 0000651d49ba14c8:0 2017-05-17 03:53:25 ServerMgr 2660 SUCCESS
Process 2660 completed Successfully

ServerLog ProcessExit 1 0000651d49ba14c8:0 2017-05-17 03:58:35 ServerMgr 3236 SUCCESS
Process 3236 completed Successfully

ServerLog ProcessCreate 1 000013f949bf1744:0 2017-05-17 03:58:48 Created server process (OS
pid = 5816 ) for ServerMgr

ServerLog ProcessExit 1 0000651d49ba14c8:0 2017-05-17 03:59:13 ServerMgr 5816 SUCCESS
Process 5816 completed Successfully

ServerLog ProcessCreate 1 000013f949bf1744:0 2017-05-17 03:59:29 Created server process (OS
pid = 5976 ) for ServerMgr

ServerLog ProcessExit 1 0000651d49ba14c8:0 2017-05-17 04:34:25 ServerMgr 7624 SUCCESS
Process 7624 completed Successfully
```

Common Event Types for Component Diagnostics

Set the event types in the following table to the indicated log levels for general server component diagnostic purposes. The increased log levels either create log files for the server component of interest or increase the amount of logging information contained in the component log files. For a description on how to set log levels for component event types, see *Setting Log Levels for Component Event Types*.

CAUTION: Increased log levels require more memory and system resources. Make sure to return the event types to their previous values after completing diagnostics.

Event Type Name	Event Type Alias	Log Level Setting
Component Tracing	Trace	4
General Events	GenericLog	4
Task Configuration	TaskConfig	4
SQL Tracing	SQL	4
SQL Error	SQLError	4
SQL Parse and Execute	SQLParseAndExecute	4

Common Event Types for Siebel Application Object Manager Diagnostics

Set the event types in the following table to the indicated log levels for general Siebel Application Object Manager diagnostic purposes. The increased log levels either create log files for the Application Object Manager of interest or increase the amount of logging information contained in the Application Object Manager component log files. Increasing the event logging provides information about the individual processes and steps that are part of the Application Object Manager task.

For a description on how to set log levels for Application Object Manager component event types, see *Setting Log Levels for Component Event Types*.

CAUTION: Increased log levels require more memory and system resources. Make sure to return the event types to their previous values after completing diagnostics.

Event Type Name	Event Type Alias	Log Level Setting	Description
Event to track the flow of a message	MessageFlow	4	Captures messages exchanged between the Siebel Application Object Manager and Siebel Application Interface.
Object Manager Session Operation and SetErrorMsg Log	ObjMgrSessionLog	4	Captures user session login, logout, and timeout information.

Event Type Name	Event Type Alias	Log Level Setting	Description
		5	Captures user name and IP address when the session completes.
Event Context	EventContext	4	Captures applet and method executed, view names, and screen names that the user navigates to.
General Object Manager Log	ObjMgrMiscLog	5	Captures general Application Object Manager events: load license, errors, and so on.
Object Manager Business Component Operation and SetErrorMsg Log	ObjMgrBusCompLog	4	Captures business component-related events: create and delete.
Object Manager Business Service Log	ObjMgrBusServiceLog	4	Captures business service-related events: create, delete, methods invoked, and so on.
Main Thread Events	MainThread	4	Captures task counter, task creates, and task exits (in main Multithreaded Server log).
Task Related Events	TaskEvents	4	Captures task creation, context, session timeout, and close info.
SQL Parse and Execute	SQLParseAndExecute	4	Captures the SQL insert, update, and delete statements processed by the database connector. It includes the SQL statement and bind variables. The content is similar to the ObjMgrSqlLog event; however, the select statement is not captured by the SQLParseAndExecute event.
Object Manager SQL Log	ObjMgrSqlLog	4	Captures the SQL select, insert, update, and delete statements processed by the Application Object Manager data object layer. Includes the SQL statement and bind variables. It also captures the preparation, execute, and fetch time for the SQL cursor.
		5	Captures internal and customer-defined search and sort specifications, the joins processed for queries, as well as a call stack of the operation performed. Setting this event to log level 5 incurs a significant performance impact because a callstack is generated. Only set this event to log level 5 in consultation with Oracle Global Customer Support.
SQL Summary	SQLSummary	4	Captures SQL prepare, fetch, and execute times. Provides detailed information regarding the execution of an SQL statement.
Security Adapter Log	SecAdptLog	5	Captures security adapter tracing information to the Application Object Manager log file.

Event Type Name	Event Type Alias	Log Level Setting	Description
Security Manager Log	SecMgrLog	5	Captures security manager tracing information to the Application Object Manager log file.

5 Configuring Additional System Logging

Configuring Additional System Logging

This chapter describes other system logging configurations and information that can be used to uncover errors or improper application behavior in addition to Siebel Server and component event logging. It includes the following topics:

About Environment Variables for System Logging

The following system environment variables can be set to assist with logging other aspects of the Siebel application deployment. For information about configuring these environment variables on both Microsoft Windows and UNIX, see *Siebel System Administration Guide* or review the documentation specific to your operating system for details on changing these variables.

- **SIEBEL_LOG_EVENTS.** The SIEBEL_LOG_EVENTS environment variable sets the event logging level, which determines the extent of information captured in the log file. For level settings and descriptions of information captured, see the table in *About Events and Event Logging*. More information is captured when the environment variable is set to a higher numeric value, and less information is captured when the variable is set to a lower numeric value. The numeric value is inversely proportional to the severity of the information (0 is more severe than 5, for instance). More disk space is consumed and performance is hindered when the value is set to a value of 5 than a value of 0.
- **SIEBEL_LOG_ARCHIVES.** The SIEBEL_LOG_ARCHIVES environment variable determines the number of log files archived. Set this value to a positive integer; this value indicates the number of files that are saved. For example, if the value is 3, then only the three most recent log files are retained, any additional log files are deleted. When a new log is created, program.log, the previous versions are archived as program_1.log, program_2.log, and so on. The numbers in the filename increase as the file becomes Statistics older. The oldest log file that numbers past the integer setting is deleted. The default value of this variable is ten.
- **SIEBEL_LOG_DIR.** The SIEBEL_LOG_DIR environment variable determines the log file location. Set this variable to change the location from the default directory. Make sure this directory already exists, access permission to write a file in that location is available, and sufficient space is free to support the log file.
- **SIEBEL_CRASH_HANDLER.** The SIEBEL_CRASH_HANDLER environment variable enables the creation of files when there is a malfunction or failure. For information about these files, see *About Other Siebel Server Log Files*. The default setting is 1, which enables the creation of such files. Setting this variable to 0 disables this function. Only set this variable in consultation with Oracle Global Customer Support. For help with setting this variable, create a service request (SR) on My Oracle Support.

- **SIEBEL_ASSERT_MODE.** The SIEBEL_ASSERT_MODE environment variable enables the creation of assert files. For information about assert files, see *About Other Siebel Server Log Files*. The default setting is 0, which disables the creation of assert files. Only set this variable in consultation with Oracle Global Customer Support. For help with setting this variable, create a service request (SR) on My Oracle Support.
- **SIEBEL_SESSMGR_TRACE.** The SIEBEL_SESSMGR_TRACE environment variable enables tracing for session manager, which is part of the Siebel Application Interface. By default, this variable is set to 0, which logs fatal and error events to the Siebel Application Interface log file. For information about Siebel Application Interface log files, see *About Logging for Siebel Application Interface*. To enable detailed logging of session manager, set this variable to 1. For more information about configuring logging for Siebel Application Interface, see *Configuring Siebel Application Interface Logging*.
- **SIEBEL_SISNAPI_TRACE.** The SIEBEL_SISNAPI_TRACE environment variable enables tracing for SISNAPI, which is a Siebel-proprietary communication protocol between the Siebel Application Interface and the Siebel Servers. By default, this variable is set to 0, which logs fatal and error events to the Siebel Application Interface log file. For information about Siebel Application Interface log files, see *About Logging for Siebel Application Interface*. To enable detailed logging of SISNAPI, set this variable to 1. For more information about configuring logging for Siebel Application Interface, see *Configuring Siebel Application Interface Logging*.
- **SIEBEL_STDERROUT.** The SIEBEL_STDERROUT environment variable enables logging of the standard error files. For more information about standard error files, see *Configuring Standard Error Files*. By default, this variable is set to 0, which disables standard error file logging. To enable logging of standard error files, set this variable to 1.

Configuring Siebel Gateway Log Files

The Siebel Gateway logs information in the cloudgateway.log file that is related to the application container in the `SIEBEL_ROOT/applicationcontainer_internal` directory where the Siebel Gateway is installed, such as `/siebel/ses/applicationcontainer_internal`. For information about installing and configuring the Siebel Gateway, see *Siebel Installation Guide* .

If you plan to configure the log level for the Siebel Gateway application container, it is recommended to do so after you finish configuring Siebel Gateway using Siebel Management Console.

Logging for the cloudgateway.log file involves editing the log4j2-siebel.properties file. (Values from the gateway.properties file are no longer used for Siebel Gateway logging.) After modifying the log4j2-siebel.properties file, you must restart the application container, as described in *Siebel Installation Guide* .

To configure Siebel Gateway logging for the cloudgateway.log file

1. Edit the log4j2-siebel.properties file, which is located in the `SIEBEL_ROOT/applicationcontainer_internal/webapps/siebel/WEB-INF/` directory where the Siebel Gateway is installed.
2. When configuring the log level, modify only the following section, and no other part of the file. The default log level configured in the file is ERROR. For example, you can change the level from ERROR to ALL.

   For log4j2-siebel.properties

```
logger.Servlet.level = ALL
logger.Servlet.additivity = false
logger.Servlet.appenderRef.rolling.ref = Servlet

logger.CG_LOGGER.level = ALL
logger.CG_LOGGER.name = CG_LOGGER
logger.CG_LOGGER.additivity = false
```

```
logger.CG_LOGGER.appenderRef.rolling.ref = CG_LOGGER
```

Set the level to one of the following values:

- **ALL.** The ALL level has the lowest possible rank and is intended to turn on all logging.
- **TRACE.** The TRACE level designates finer-grained informational events than DEBUG.
- **DEBUG.** The DEBUG level designates fine-grained informational events that are most useful to debug an application.
- **INFO.** The INFO level designates informational messages that highlight the progress of the application at coarse-grained level.
- **WARN.** The WARN level designates potentially harmful situations.
- **ERROR.** The ERROR level designates error events that might still allow the application to continue running.
- **FATAL.** The FATAL level designates very severe error events that will presumably lead the application to abort.

Additionally, you can use the following log level to turn off logging:

- **OFF.** The OFF level has the highest possible rank and is intended to turn off logging.

3. Optionally, you can configure the log4j2-siebel.properties file to include the `monitorInterval` parameter, which allows changing the log level at runtime for all application container logs that are controlled by this file and not by profiles in Siebel Management Console. If you change the following line:

For log4j2-siebel.properties

```
status = warn
name = Siebel
monitorinterval = 300
```

Then Apache Tomcat checks for changes in the log level within the specified interval (300 seconds, or 5 minutes) and, if the log level changed, then the log statements are written to log files to reflect the change in the log level.

Related Topics

Configuring the Common Logger for Siebel Application Interface

Related Books

Siebel Installation Guide

Configuring Standard Error Files

Standard error files contain process messages that are directed to standard error and standard out. These messages come from Siebel Server or third-party components and contain important information to help diagnose Siebel Server functionality issues. For example, the information contained in a Siebel Server process message can help identify instances where `siebmtshmw`, the process shell in which the Siebel Application Object Manager component runs, is

unable to start due to problems like incorrect LIBPATH setting or a corrupt registry. For more information about Siebel Server processes, see *Siebel System Administration Guide* .

When configured, process messages are saved to file in the directory labeled `SIEBSRVR_ROOT/log/StdErrOut`. The format of the standard error files is as follows:

```
stderrout_$Process_ID_$Timestamp.log
```

where:

- *Process_ID* is the operating system process ID number (PID).
- *Timestamp* is the log file creation time, in YYYY-MM-DD HH:MM:SS format.

Standard error file logging is not enabled by default.

To configure standard error file logging

1. On the computer running the Siebel Server, set the following environment variable to the given value:
    - SIEBEL_STDERROUT = 1

    For more information about this variable, see *About Environment Variables for System Logging*.

    For more information about setting environment variables, see *Siebel System Administration Guide* .
2. Stop and restart the computer running the Siebel Server for the environment variable to take effect.

About Other Siebel Server Log Files

Siebel CRM generates other text log files in the binary (`bin`) subdirectory of the Siebel Server root directory. These files record conditional responses when certain portions of code are executed during the operation of the application. They appear in the following form listed in the following table.

Log Filename	Description
siebel_assert*.txt	Indicates a fatal condition that might have led to a failure or data corruption.
siebel_crash*.txt	Indicates a process has malfunctioned or failed. These files are produced on UNIX platforms only.
siebel_prefer*.txt	Indicate a less critical error condition that arises but did not lead to a failure, malfunction, or data corruption.

If these files are generated during the normal running of processes when no errors occur, then they can be ignored (or deleted, because they can become very large). However, if these files are generated when errors occur (especially failures), then you can send these files to Oracle Global Customer Support for investigation by creating a service request (SR) on My Oracle Support.

About Flight Data Recorder Log Files

Siebel flight data recorder (FDR) files are records of system and server component behavior at run time. In the event of a system or server component failure, the settings and events leading up to the failure are captured and logged. You can send the Siebel flight data recorder log file to Oracle Global Customer Support by creating a service request (SR) on My Oracle Support. The log file is used to troubleshoot and analyze the specific settings and events that occurred before the failure.

The Siebel flight data recorder log files are stored in the binary (`bin`) subdirectory of the Siebel Server root directory. They appear in the following form:

```
TYYYYMMDDHHMM_PProcess_ID.fdr
```

where:

- *YYYYMMDDHHMM* is the timestamp.
- *Process_ID* is the identification number of the process that failed or was stopped.

For example:

T201705181601_P001376.fdr

is a filename that is based on a component that was started on May 18, 2017 at 4:01 PM, where the process ID value was 1376.

The Siebel flight data recorder feature is enabled by default. However, FDR activation requires the execution of at least one instrumentation point to generate a log file. If a failure happens before execution of the first instrumentation point, then no log file is generated. Instrumentation points are embedded in some workflow business services to provide capture-processing details in case of a system failure or server component failure. For more information about instrumentation and instrumentation points, see *Siebel Performance Tuning Guide* and *Siebel Business Process Framework: Workflow Guide* , respectively.

Note: FDR files are stored in binary format and cannot be read with a text editor.

Setting the environment variable SIEBEL_CRASH_HANDLER to 0 disables the creation of FDR files, in addition to several other logging functions. Only set this variable to 0 in consultation with Oracle Siebel Global Customer Support by creating a service request (SR) on My Oracle Support.

About Java EE Connector Architecture Logging

The Java EE Connector Architecture (JCA) provides a Java interface solution between application servers and Enterprise Information Systems (EIS). Siebel CRM supports JCA with the Siebel Resource Adapter. The Siebel Resource Adapter supports the invocation of business services to perform operations, such as pooling connections and managing security. JCA allows you to keep logs for such operations. For more information about JCA logging, see *Transports and Interfaces: Siebel Enterprise Application Integration* . For more information about JCA, see: http://www.oracle.com/technetwork/java/index.html

6 Querying System Log Files

Querying System Log Files

Querying log files produced by a Siebel application is a useful diagnostic task to resolve problems that occur during any stage of operation. The Log File Analyzer (LFA) is a command-line utility that assists with this analysis. It includes the following topics:

About the Log File Analyzer

The Siebel Log File Analyzer (LFA) is a command-line utility designed to search through Siebel log files and isolate information of interest. Use the LFA to analyze and review the content of log files and to compile analysis information from these files.

Run the LFA to query log files across Siebel Servers and Siebel Application Interface while filtering on one or more of the following items:

- User name
- Log levels

- Literal values
- Events or subevents
- Session IDs
- Time and date of log files
- Component

The LFA creates analysis output, which can be reviewed from the command-line or saved to text files.

For details on the process to run the LFA, see *Process for Analyzing Log Files with LFA*.

LFA Language Considerations

The LFA uses information in the events of the main Siebel Server log file to determine what components are available. The events in this log file are translated for different languages. To understand the format of the events for different languages, the LFA reads information in the language files located in the `locale` subdirectory of the Siebel Server root directory (for example, `/siebel/siebsrvr/locale`).

If the language files are changed, then the LFA might not be able to recognize certain key events in the main Siebel Server log file, which lead to run-time errors.

Strategy for Analyzing Log Files

The strategy for analyzing log files depends on the type of issues encountered. Identify whether the issue of interest is related to a particular user or the application system in general. Run the Log File Analyzer (LFA) using the strategy applicable to the identified issue.

- For a strategy to use the LFA to examine user issues, see *Analyzing User Issues*

  .
- For a strategy to use the LFA to examine system issues, see *Analyzing System Issues*

  .

For information and details on the process of using the LFA, see *Process for Analyzing Log Files with LFA*.

Analyzing User Issues

For user issues that are not immediately resolvable, log files provide additional information logged by the application regarding a user's time spent accessing and using the application.

The LFA gives the administrator the capability of querying across numerous log files for log events that were pertinent to the user's session. For example, in a situation where a user named Casey Smith reports an issue with her application at approximately 13:00, use the LFA to query events pertinent to Casey that occurred between 12:30 and 14:00. To refine the results, include the condition that the log level must be greater than or equal to one, which represents an error condition.

The LFA output includes information as to which file each log event came from. The administrator can, after finding an error or other log event of interest, check back in the original log file and look for events nearby that might give additional context useful for troubleshooting the issue.

Note: To query log files for users, make sure the environment variable SIEBEL_LOG_EVENTS is set to 4. For more information about environment variables, see *Common Event Types for Component Diagnostics*.

Analyzing System Issues

For general system issues not involving user issues (for example, a problem with a workflow), the LFA assists the administrator in isolating and resolving issues relating to general system usage.

For example, if the workflow processor is known to have failed within a particular time frame, then use the LFA to search for log events that occurred during that time frame, and then look at the log files in which the events are contained for more specific detail.

As a preventative measure, the LFA is also useful to periodically check log files for any errors, even if no system issue is apparent at that time.

Process for Analyzing Log Files with LFA

To analyze log files with the Log File Analyzer (LFA), perform the following tasks:

1. Configure the LFA to access the appropriate Siebel Server and Siebel Application Interface log files, if necessary. For more information about this task, see *Configuring the Log File Analyzer*.
2. Start the LFA. For more information about this task, see *Starting the Log File Analyzer*.
3. Query the log files using LFA. For information about this task, see *Creating and Saving LFA Queries*. For general information about running the LFA, see *About Running Log File Analyzer Commands*.

For strategies on analyzing log files using the LFA, see *Strategy for Analyzing Log Files*.

Configuring the Log File Analyzer

Configure the Log File Analyzer (LFA) by accessing and editing the LFA configuration file, which has the default name, logreader.cfg. The LFA uses the LFA configuration file when started to reference Siebel Server locations, Siebel Application Interface locations, and other run-time details.

This task is the first step in *Process for Analyzing Log Files with LFA*. Once the LFA is configured, this step is optional unless further changes are necessary.

The default location for the LFA configuration file is the binary (`bin`) subdirectory of the Siebel Server root directory (for example, `/siebel/siebsrvr/bin`).

The LFA configuration file contains sections that configure which log files are analyzed by the utility and what content is reviewed. Edit the appropriate sections in the configuration file with a text editor. For LFA configuration file parameters and their descriptions, see the following table. For an example of a typical configuration file, see *Example of a Log File Analyzer Configuration File*.

Section	Parameter	Description

[elements]	Siebel Server Identification Tag	Under the [elements] section, list Siebel Servers searchable by the LFA using the following format:
		
		`Siebel Server Identification Tag = server`
		
		where Siebel Server Identification Tag is a unique tag name identifying the Siebel Server of interest.
		
		This tag can be the Siebel Server name, but can also be any other configurable value, for example:
		
		`[elements]`
		`SiebelServer1=server`

	Siebel Application Interface Identification Tag	Under the [elements] section, list Siebel Application Interface instances that are searchable by the LFA using the following format:
		
		`Siebel Application Interface Identification Tag = plug-in`
		
		where Siebel Application Interface Identification Tag is a unique tag identifying the Siebel Application Interface of interest.
		
		This tag can be the Siebel Application Interface name, but can also be any other configurable value, for example:
		
		`[elements]`
		`SiebelAI1=plugin`

[Siebel Server Identification Tag]	Path	Each Siebel Server identification parameter listed in the [elements] section has a respective section of its own with its name in square brackets. The path parameter of each Siebel Server section denotes the location of the associated log files for that Siebel Server. For example:
		
		`[SiebelServer1]`
		`Path = //siebel/ses/siebsrvr/log`

[Siebel Server Identification Tag.Siebel Server Component Name]	shortname	List Siebel Server component display names in square brackets to allow the LFA to search for component references in log files specific to a Siebel Server. Add the Siebel Server component alias as the value for the short name parameter. For example:
		
		`[SiebelServer1.Server Request Broker]`
		`shortName=SRBroker`
		
		For a listing of Siebel Server components and their aliases, see *Siebel System Administration Guide* .

[Siebel Application Interface Identification Tag]	Path	Each Siebel Application Interface identification parameter listed in the [elements] section has a section of its own with its name in brackets ([ ]). The path parameter of each Siebel Application Interface section denotes the location of the associated log files for that Siebel Application Interface. For example:
		
		`[SiebelAI1]`
		`Path = //siebel/SiebelAI1/log`

[Render]	event	Displays information on log events, if enabled. Set to 1 to enable; set to 0 to disable.

Section	Parameter	Description
		The parameter information in the [render] section is also controlled by using commands during the running of the LFA. For more information, see *About Running Log File Analyzer Commands*.
	subevent	Displays information on log subevents, if enabled. Set to 1 to enable; set to 0 to disable.
	loglevel	Displays information on log level of event subtypes. Set to 1 to enable; set to 0 to disable.
	time	Displays log timing information in enabled. Set to 1 to enable; set to 0 to disable.
	file	Displays log file path information, if enabled. Set to 1 to enable; set to 0 to disable.

Note: Do not modify the sections entitled [schemes], [user], and [session].

Example of a Log File Analyzer Configuration File

The following example Log File Analyzer (LFA) configuration file is intended for a Siebel application with two Siebel Servers, named SiebSrv1 and SiebSrv2, and three instances of Siebel Application Interface, named SiebelAI1, SiebelAI2, and SiebelAI3. The LFA configuration file also contains alias information on two Siebel Server components, Server Request Broker and Call Center Object Manager. Using this configuration file, the LFA searches all Siebel Server and Siebel Application Interface log files, has the ability to search on the two Siebel Server components listed, and displays all information except log level and the log file path.

For descriptions of the individual sections and parameters, see *Configuring the Log File Analyzer*.

```
[elements]
SiebSrv1=server
SiebSrv2=server
SiebelAI1=plugin
SiebelAI2=plugin
SiebelAI3=plugin

[SiebSrv1]
Path = //siebel/ses/siebsrvr/log

[SiebSrv2]
Path = //siebel/ses/siebsrvr/log

[SiebSrv1.Server Request Broker]
shortName=SRBroker

[SiebSrv2.Call Center Object Manager (ENU)]
shortName=SCCObjMgr

[SiebelAI1]
Path = //siebel/SiebelAI1/log

[SiebelAI2]
Path = //siebel/SiebelAI2/log

[SiebelAI3]
```

```
Path = //siebel/SiebelAI3/log

[Render]
event=1
subevent=1
loglevel=0
time=1
file = 0
```

Starting the Log File Analyzer

Starting the Log File Analyzer (LFA) is the second step in the *Process for Analyzing Log Files with LFA*. For background information about the LFA, see *About the Log File Analyzer*.

The LFA utility resides in the binary (`bin`) subdirectory of Siebel Server root directory on Microsoft Windows as the executable logreader.exe or as binaries on UNIX.

The procedure for starting the LFA on Microsoft Windows is available in *Starting the Log File Analyzer on Microsoft Windows*.

The procedure for starting the LFA on UNIX is available in *Starting the Log File Analyzer on UNIX*.

Starting the Log File Analyzer on Microsoft Windows

Use the following command to start the Log File Analyzer (LFA) command-line utility on Microsoft Windows.

To start the Log File Analyzer on Microsoft Windows

1. Navigate to the binary (`bin`) subdirectory within the Siebel Server root directory (for example, `C:\siebel\siebsrvr\bin`).
2. Make sure the LFA configuration file (logreader.cfg) is present in the same directory as the utility. If this file is located in another directory, or has another name, then use the `/f` parameter described in the following table. For more information about the configuration file, see *Configuring the Log File Analyzer*.
3. At the Windows command prompt, enter logreader.exe using, as necessary, parameters listed in the following table.
   The log reader command prompt appears after a successful start as follows:
   ```
   logreader>
   ```
4. Run the LFA by using the commands described in *About Running Log File Analyzer Commands*.

The following table describes the parameters available for use during the starting of the LFA.

Parameter	Description	Example
/h	Lists the parameters available for use with the LFA utility.	`logreader /h`
/f	Locates the LFA configuration file if it is not present in the LFA utility directory or if the configuration file is named differently than logreader.cfg. Include the path or new configuration filename after the `/`	`logreader /f abc.cfg` or

Parameter	Description	Example
	`f` parameter. If the configuration filename includes a space, then enclose the argument with quotation marks.	`log reader /f g:\abc\abc.cfg`
/i	Specifies an input file that contains LFA commands. At startup, the LFA provides output from the commands listed in the input file. Include the filename and path, if necessary, after the `/i` parameter.	`logreader /i g:\abc\abc.txt`

Note: Use the `/f` and `/i` parameters independently or together.

Starting the Log File Analyzer on UNIX

Use the following command to start the Log File Analyzer (LFA) command-line utility on UNIX operating systems.

To start the Log File Analyzer on UNIX

1. Make sure the LIBPATH (AIX) environment variable contains the full pathname for your database client library directory. For more information about these variables, see *Siebel Installation Guide* .
2. Make sure the LFA configuration file (logreader.cfg) is present in the same directory as the utility. If this file is located in another directory, or has another name, then use the `/f` parameter described in the table in *Starting the Log File Analyzer on Microsoft Windows*. For more information about the configuration file, see *Configuring the Log File Analyzer*.
3. Enter `logreader` using, as necessary, other parameters listed in the table in *Starting the Log File Analyzer on Microsoft Windows*.
   The logreader command prompt appears after a successful start as follows:
   ```
   logreader>
   ```
4. Run the LFA by using the commands described in *About Running Log File Analyzer Commands*.

About Running Log File Analyzer Commands

Running the Log File Analyzer (LFA) allows you to search and filter information contained in Siebel application log files. For overall strategy on running the LFA, see *Strategy for Analyzing Log Files*.

Make sure when running the LFA that you enter commands and parameters correctly. The following information is common to all LFA commands:

- The LFA is case sensitive.
- Enclose any parameters that contain spaces with quotation marks.

The following topics list instructions for running the LFA:

- *Creating and Saving LFA Queries*. Creating and executing a query is the fundamental task associated with the LFA.
- *Filtering LFA Queries*. Filtering queries assists the user to isolate diagnostic information of interest.

Note: Move log files to a nonproduction environment before querying them with the LFA. As the LFA parses through potentially large and numerous log files, using the LFA in a production environment might reduce overall system performance.

Creating and Saving LFA Queries

Creating and executing a query is the fundamental task associated with the Log File Analyzer (LFA). Creating saved queries is a task in the *Process for Analyzing Log Files with LFA*.

Run queries using the LFA `query` command to search log files based on users, literal values, sessions, severity, events, subevents, log times, or combinations of these items.

For descriptions on running these commands, see the following topics.

The LFA saves the results of each query to memory or saves it to a text file. For details on displaying saved queries, see *Displaying Saved Query Output*. For details on saving output to a text file, see *Saving Log File Analyzer Output to Text Files*.

To stop a query before it finishes, see *Interrupting Log File Analyzer Queries*.

Querying Log Files for Users

To query log files for users, you must first set an environment variable log level, and add sections to the logreader.cfg file.

Note: To run queries for a specific user using the option *user*, you must run the query against the Siebel Application Interface log files.

Use the following procedures to query log files for users.

To prepare for querying events associated with users

1. Make sure the SIEBEL_LOG_EVENTS environment variable is set to 4.

   For more information about setting environment variables, see *Setting Log Levels for Component Event Types*.
2. Add new sections to the logreader.cfg file by doing the following:
   a. Add a SWE=plugin section under the [elements] section.
   b. Create a new [SWE] section with the following:

      Path = `<swe log files directory location>`.

Use the following procedure to search log files for events associated with individual users.

To query for events associated with a particular user

- Using the command-line interface, enter:

```
query query_name where user = user_name
```

where:

- query_name is the query command output stored in memory under this name.
- user_name is the user of interest in log files.

An example of this query command is as follows:

```
query asqry where user = asmith
```

This command queries log files for events associated with user `asmith` and saves the output to memory under the name `asqry`.

For other options of the Log File Analyzer (LFA) query command, see *Creating and Saving LFA Queries*.

Querying Log Files for Literal Values

Use the following procedure to search log files for specific literal values. For other options of the Log File Analyzer (LFA) query command, see *Creating and Saving LFA Queries*.

To query for a literal value

- Enter:

```
query query_name where literal = literal_value
```

where:

- query_name is the query command output stored in memory under this name.
- literal_value is the literal value of interest in log files.

An example of this query command is as follows:

```
query litqry where literal = Parameter
```

This command queries log files for events associated with literal Parameter and saves the output to memory under the name `litqry`.

Querying Log Files for Error Messages

Use the following procedure to search log files for error messages. This command is an application of querying for literal values. For other options of the Log File Analyzer (LFA) query command, see *Creating and Saving LFA Queries*.

To query for an error message

- Enter:

  ```
  query query_name where literal = error_message_number
  ```

  where:

  - query_name is the query command output stored in memory under this name.
  - error_message_number is the error message number of interest in log files.

An example of this query command is as follows:

```
query errorqry where literal = SBL-ASG-00001
```

This command queries log files for events associated with error message number SBL-ASG-00001 and saves the output to memory under the name `errorqry`.

Querying Log Files for Sessions

Use the following procedure to search log files for specific sessions. For other options of the Log File Analyzer (LFA) query command, see *Creating and Saving LFA Queries*.

To query for events associated with a particular session

- Enter:

  ```
  query query_name where session = session_ID
  ```

  where:

  - query_name is the query command output stored in memory under this name.
  - session_ID is the session ID of interest in log files.

An example of this query command is as follows:

```
query sesqry where session = !1.15bc.c425.3f302b17
```

This command queries log files for events associated with session ID `!1.15bc.c425.3f302b17` and saves the output to memory under the name `sesqry`.

Querying Log Files of a Particular Severity

Use the following procedure to search log files for events of a specific severity. For other options of the Log File Analyzer (LFA) query command, see *Creating and Saving LFA Queries*.

Events are categorized from 0 to 5, 0 being the most severe or critical. For more information about event severity and event logging, see *Siebel System Administration Guide* .

This command includes events of the indicated severity as well as events of a greater severity. For example, if you query for a severity of 2, then events of severity 0 and 1 are also included in the output.

To query for events associated with a particular severity

- Enter:

  ```
  query query_name where loglevel = severity_value
  ```

  where:

  - query_name is the query command output stored in memory under this name.
  - severity_value is the severity value of interest (integer value from 0 to 5).

An example of this query command is as follows:

```
query svtqry where loglevel = 1
```

This command queries log files for events associated with a severity of 0 and 1 and saves the output to memory under the name `svtqry`.

Querying Log Files for a Particular Log Event

Use the following procedure to search log files for a specific log event. For other options of the Log File Analyzer (LFA) query command, see *Creating and Saving LFA Queries*.

For a partial listing of log events and for more information about event logging, see *Siebel System Administration Guide* .

To query for events associated with a particular log event

- Enter:

  ```
  query query_name where event = event_name
  ```

  where:

  - query_name is the query command output stored in memory under this name.
  - event_name is the log event name of interest.

An example of this query command is as follows:

```
query evtqry where event = SessMgr
```

This command queries log files for log events named `SessMgr` and saves the output to memory under the name `evtqry`.

Querying Log Files with a Particular Log Subevent

Use the following procedure to search log files for a specific log subevent. For other options of the Log File Analyzer (LFA) query command, see *Creating and Saving LFA Queries*.

For a partial listing of log subevents and for more information about event logging, see *Siebel System Administration Guide* .

To query log entries associated with a particular log subevent

- Enter:

  ```
  query query_name where subevent = subevent_name
  ```

  where:

  - query_name is the query command output stored in memory under this name.
  - subevent_name is the log subevent name of interest.

An example of this query command is as follows:

```
query subevtqry where subevent = SisnNetGeneric
```

This command queries log files for log subevents named `SisnNetGeneric` and saves the output to memory under the name `subevtqry`.

Querying Log Files After a Particular Time

Use the following procedure to search log files created after a specific time. For other options of the Log File Analyzer (LFA) query command, see *Creating and Saving LFA Queries*.

To query events logged after a certain time

- Enter:

  ```
  query query_name where time from "YYYY-MM-DD
   HH:MM:SS"
  ```

  where:

  - query_name is the query command output stored in memory under this name.
  - "YYYY-MM-DD HH:MM:SS" is the date and time of interest.

  > **Note:** The exact time portion of the date and time parameter, `HH:MM:SS`, can be omitted. In this case, the date's base time defaults to `00:00:00`.

An example of this query command is as follows:

```
query timeqry where time from "2017-05-01 16:30:00"
```

This command queries log files created after May 1, 2017 at 4:30 PM, and saves the output to memory under the name `timeqry`.

This command is useful in combination with other parameters to filter results. For more information, see *Querying Log Files Using Multiple Conditions*.

Querying Log Files Within a Time Interval

Use the following procedure to search log files created within a specific time interval. For other options of the Log File Analyzer (LFA) query command, see *Creating and Saving LFA Queries*.

To query events logged within a certain time interval

- Enter:

  ```
  query query_name where time from "YYYY-MM-DD
   HH:MM:SS" to "YYYY-MM-DD
   HH:MM:SS"
  ```

  where:

  - *query_name* is the query command output stored in memory under this name.
  - "*YYYY-MM-DD HH:MM:SS*" is the date and time of interest.

  > **Note:** The exact time portion of the date and time parameter, `HH:MM:SS`, can be omitted. In this case, the date's from-time defaults to `00:00:00` and the to-time defaults to `23:59:59`.

An example of this query command is as follows:

```
query timeintqry where time from "2017-05-01 16:30:00" to "2017-05-05"
```

This command queries log files created between May 1, 2017 at 4:30 PM and May 5, 2017 at 11:59 PM, and saves the output to memory under the name `timeintqry`.

This command is useful in combination with other parameters to filter results. For more information, see *Querying Log Files Using Multiple Conditions*.

Querying Log Files for Components

Use the following procedure to search log files for a specific Siebel Server component. For other options of the Log File Analyzer (LFA) query command, see *Creating and Saving LFA Queries*.

Make sure the LFA configuration file contains information on the Siebel Server component of interest. For more information, see *Configuring the Log File Analyzer*.

For more information about Siebel Server components, see *Siebel System Administration Guide* .

To query log entries for a particular Siebel Server component

- Enter:

  ```
  query query_name where component = component_name
  ```

  where:

  - query_name is the query command output stored in memory under this name.
  - component_name is the Siebel Server component name of interest.

Note: The component_name parameter value is either the long form or alias form of the Siebel Server component name. For a list of component names and aliases, see *Siebel System Administration Guide* .

An example of this query command is as follows:

```
query compqry where component = SCCObjMgr
```

This command queries log files for the Call Center Object Manager (alias SCCObjMgr) and saves the output to memory under the name `compqry`.

Querying Log Files Using Multiple Conditions

This topic provides examples of combination query commands using multiple conditions. For a list of individual query command conditions and their use, see *Creating and Saving LFA Queries*.

The logical AND and OR operators are also applicable to the Log File Analyzer (LFA) query command. To add clarity to multiple condition commands, group condition sets in parentheses.

- `query litasqry where (literal = Parameter) or (user = asmith)`

  This command queries log files for the literal `Parameter` or the user `asmith`. It saves the output to memory under the name `litasqry`.

- `query aqry where literal = Parameter and literal = SBL-GEN`

  This command queries log files for the literal `Parameter` and the literal `SBL-GEN`. It saves the output to memory under the name `aqry`.

- `query asaugqry where user = asmith time from 2017-05-05`

  This command queries log files for the user `asmith` after May 05, 2017. It saves the output to memory under the name `asaugqry`.

- `query asaugqry where user = asmith time from "2017-05-05 15:20:00" to "2017-05-05 15:30:00"`

  This command queries log files for the user `asmith` during the ten-minute period between 3:20 PM and 3:30 PM on May 05, 2017. It saves the output to memory under the name `asaugqry`.

Filtering LFA Queries

Use the **show** command to further refine the output of saved queries. For information about querying log files and creating saved queries, see *Creating and Saving LFA Queries*.

For information about displaying a saved query or multiple saved queries, see *Displaying Saved Query Output*.

To filter saved query information

- Enter:

  ```
  show query_name
   where_clause
  ```

  where:

  - query_name is the query command output stored in memory under this name.
  - where_clause is the WHERE clause used to filter display results using key words.

For a list of key words available for use with the Log File Analyzer (LFA), see *Listing Query Command Key Words*. The syntax of where clauses used with the **show** command are similar to those used with the **query** commands. Review *Creating and Saving LFA Queries* for more information.

Use multiple where clause conditions and the logical operators AND and OR to further filter an individual or multiple saved queries. For examples of these types of commands, see *Examples of Filtered Saved Queries*.

To save filtered output from the **show** command, save the results to a text file. For description of this task, see *Saving Log File Analyzer Output to Text Files*. Filtered output from the **show** command cannot be saved in memory.

Examples of Filtered Saved Queries

The following examples display the type of filtering available on saved queries using the **show** command.

- **show aquery where user = asmith**

  This command filters the saved query **aquery** for information specific to user **asmith**.

- **show aquery where user = asmith and literal = Parameter time from "2017-05-05 15:20:20" to "2017-05-05 15:30:00" > out.dat**

  This command filters the saved query **aquery** for information on user **asmith** and the literal value **parameter** between the time of 3:20 and 3:30 PM on May 05, 2017. The command also stores the results of the filtered query to a text file named out.dat.

- **show aquery, bquery where user = asmith and literal = Parameter time from "2017-05-05 15:20:20" to "2017-05-05 15:30:00" > out.dat**

  This command filters the saved queries **aquery** and **bquery** based on the same conditions in the previous bullet.

Saving Log File Analyzer Output to Text Files

Use the following procedure to save the results of a Log File Analyzer (LFA) command to a text file. For information about running the LFA, see *About Running Log File Analyzer Commands*. Any LFA command that creates output can have the output channeled to a file.

To save Log File Analyzer output to text files

- Enter:

```
log_file_analyzer_command
 >
file_name.txt
```

  where:
  - log_file_analyzer_command is the LFA command.
  - file_name.txt is the name of the output text file.
    Make sure to:
  - Include the > character when saving output to a text file.
  - Specify a path name with the text filename if you want to save the log file to another directory, and not the Log File Analyzer (LFA) directory.

Example:

```
query litqry where literal = Parameter > output1.txt
```

This command saves the output from the `litqry` saved query to the text file named `output1.txt`. The LFA stores this output text file in the same directory as the Log File Analyzer directory.

Displaying Saved Query Output

Use the following procedures to display results of one or more saved query commands to the screen. For a listing of saved queries, see *Listing Log File Analyzer Queries and Run-time Details*.

For more information about the query command, see *Creating and Saving LFA Queries*.

The Log File Analyzer (LFA) also saves query command output to text files. For more information about this task, see *Saving Log File Analyzer Output to Text Files*.

To show saved query output to the screen

- Enter:

```
show query_name
```

In this command, query_name is the query command output stored in memory under this name.

Example:

```
show evtqry
```

This example displays the output from a previous query command named `evtqry`.

Note: The LFA only displays queries saved to memory during a given session.

To show multiple saved query output to the screen

- Enter:

  ```
  show query_name_1, query_name_2, ... , query_name_N
  ```

  In this command, query_name_N is the query command output stored in memory under this name.

Example:

```
show evtqry1, evtqry2
```

This example displays the output from two previous query commands named `evtqry1` and `evtqry2`.

Interrupting Log File Analyzer Queries

Use the following procedure to interrupt a query command. For more information about the query command, see *Creating and Saving LFA Queries*.

To interrupt a query command in operation

- Press CTRL-C during the operation of the command.

Listing Query Command Key Words

Use the following procedure to list the key words available for use with the `query` command `where` clause. For detailed descriptions of use for each key word, see *Creating and Saving LFA Queries*.

To list the query command key words

- Enter:

  ```
  keys
  ```

The key words are output to the screen.

Listing Log Event Fields Display Status

Use the following procedure to list the display status for log event fields. The value 1 indicates the log event field is set to display. The value 0 indicates the log event field is set to hide.

To list log event fields display status

- Enter:

  ```
  fields
  ```

To change the display status at run-time, see the task *Showing Log Event Fields in LFA Results* or *Hiding Log Event Fields in LFA Results* for more information.

Set the default display status of the event log fields by modifying the Log File Analyzer (LFA) configuration file. For more information about the LFA configuration file, see *Configuring the Log File Analyzer*.

Showing Log Event Fields in LFA Results

Use the following procedures to show log file fields in the output from the Log File Analyzer (LFA) during an individual LFA session. You can also set this information in the LFA configuration file, which is applicable to all LFA sessions. For more information, see *Configuring the Log File Analyzer*.

To list the current event log field display status, see *Listing Log Event Fields Display Status*.

To show log file fields in the LFA output

- Enter:

  ```
  showfield log_field_name
  ```

  In this command, log_field_name is the name of the log field name for display.

For a list of the available display fields, see the following table.

Set multiple log file fields to show on a single `showfield` command by separating each log file field with a space or comma.

Log File Field	Description
event	Name of the event.

Log File Field	Description
subevent	Name of the subevent.
loglevel	Severity of the log file event.
file	File and path name of the log file.
time	Date and time of the log file.

Hiding Log Event Fields in LFA Results

Use the following procedures to hide log file fields in the output from the Log File Analyzer (LFA) during an individual LFA session. You can also set this information in the LFA configuration file, which is applicable to all LFA sessions. For more information, see *Configuring the Log File Analyzer*.

To list the current event log field display status, see *Listing Log Event Fields Display Status*.

To hide log file fields in the LFA output

- Enter:

  ```
  hidefield log_field_name
  ```

  In this command, log_field_name is the name of the log field name for display. For a list of the available display fields, see the table in *Showing Log Event Fields in LFA Results*.

Set multiple log file fields to hide on a single `showfield` command by separating each log file field with a space or comma.

Deleting Log File Analyzer Saved Query Results

Use the following procedure to delete saved queries. For more information about querying log files, see *Creating and Saving LFA Queries*.

> **Note:** Deleting saved queries does not delete queries saved as text files.

To delete Log File Analyzer query results

- Enter:

  ```
  delete query_name
  ```

In this command, query_name is the query command output stored in memory under this name.

Delete multiple saved queries by separating each query name with a space or comma when using the `delete` command.

Listing Log File Analyzer Queries and Run-time Details

Use the `list` command in the following procedure to list saved queries and run-time details to the screen. For information about running the Log File Analyzer (LFA), see *About Running Log File Analyzer Commands*. For information about creating saved queries, see *Creating and Saving LFA Queries*.

For information about each list item, see *Listing Log File Information Using Log File Analyzer* for details.

To list Log File Analyzer queries and run-time details

- Enter:

```
list list_item
```

In this command, list_item is the list item of interest.

For items available for listing, see the following table.

Item	Description
all	Lists all LFA items available for listing. **Note:** The LFA does not list users or sessions until you perform at least one user query.
queries	Lists LFA queries saved in the current session.
servers	Lists servers searched by LFA.
sessions	Lists sessions found in the log files searched by LFA.
plugins	Lists instances of Siebel Application Interface searched by LFA.
components	Lists components with information in log files searched by LFA.
processes	Lists processes with information in log files searched by LFA.
users	Lists users with information in the log files searched by LFA.

Note: If the LFA is not searching the appropriate server or Siebel Application Interface, then see *Configuring the Log File Analyzer* for details on configuring the LFA to search the server and Siebel Application Interface of interest.

Listing Log File Information Using Log File Analyzer

Use the `info` command in the following procedure to list detailed information on the values of the run-time details. For a list of items available for use with the info command, see *Listing Log File Analyzer Queries and Run-time Details*.

For information about running the Log File Analyzer (LFA), see *About Running Log File Analyzer Commands*. For information about creating saved queries, see *Creating and Saving LFA Queries*.

To list information on values for Log File Analyzer run-time details

- Enter:

  ```
  info info_item
  ```

  In this command, info_item is the value of a list item of interest.

For items available for listing (with the exception of list item `all` and `queries`), see the table in *Listing Log File Analyzer Queries and Run-time Details*.

List information on multiple list values by separating values with a comma or space for the info_item parameter.

For example, using the `list` command for users revealed an entry named `asmith`. Use the following command to list information on `asmith`:

```
info asmith
```

Exiting Log File Analyzer

Use the following command to exit the log file analyzer. Exiting the log file analyzer deletes saved queries for that session unless query output is saved to text files. For information about this task, see *Saving Log File Analyzer Output to Text Files*.

To exit the Log File Analyzer

- Enter:

  ```
  exit
  ```

Troubleshooting Log File Analyzer Errors

This topic provides guidelines for resolving errors and problems that the Log File Analyzer (LFA) might generate during processing. To resolve the problem, look for it in the Symptom/Error Code and Message column in the following table.

System or Error Code and Message	Diagnostic Steps or Cause	Solution
`SBL-LFA-00100 Section [%s] in configuration file is empty.`	The section indicated in the error message is blank. LFA requires content for this section.	For the correct specification of the configuration file, see *Configuring the Log File Analyzer*.
`SBL-LFA-00101 Rule "%s" appears in the configuration file but is not registered.`	A rule has been added to the LFA configuration file but not registered with the utility. Therefore, the rule is not recognized.	At this time, it is not possible to create customized rules for the LFA. Remove this rule from the configuration file.
`SBL-LFA-00102 Cannot find section [%s] in the configuration file.`	Though it is a required section, the section of the LFA configuration file indicated in the error message text is missing.	For the correct specification of the configuration file, see *Configuring the Log File Analyzer*.
`SBL-LFA-00103 There is a format problem in section [%s] of the configuration file.`	There is a formatting error in the LFA configuration file section indicated in the error message text.	For the correct specification of the configuration file, see *Configuring the Log File Analyzer*.
`SBL-LFA-00104 Value "%s" in the section is invalid or missing.`	There is a missing value in the LFA configuration file section indicated in the error message text.	For the correct specification of the configuration file, see *Configuring the Log File Analyzer*.
`SBL-LFA-00105 Time filters are invalid or have contradictory values.`	The time filter you are trying to use in your query is invalid. It is possible that the To time is before the From time.	For information about using time filters correctly, see *Querying Log Files Within a Time Interval*.
`SBL-LFA-00106 Value or Name for "%s" is a negative number.`	This value is not expected to be negative.	Provide a positive value.
`SBL-LFA-00107 Cannot open file: "%s".`	The LFA cannot write output to the given file.	Check your permissions to the file and directory. Make sure the file is not read only.
`SBL-LFA-00108 File "%s" is already in use.`	This file might be locked by another running application.	Shut down applications that might be accessing the file and try again.
`SBL-LFA-00109 Cannot create pipe for command \"%s\".`	Pipe is not supported.	This functionality is not supported.

System or Error Code and Message	Diagnostic Steps or Cause	Solution
`SBL-LFA-00110` `OUT OF MEMORY !!!!!!`	The computer on which you are using the LFA has run out of memory.	Shut down some of your applications and try again.
`SBL-LFA-00112` `Query's "where" clause is invalid.`	The `where` clause in the query is not correctly specified.	For information about correct application of the WHERE clause, see *Creating and Saving LFA Queries*.
`SBL-LFA-00113` `Query with name "%s" does not exist.`	You have tried to reference a query that does not exist.	Type `list queries` to see existing queries. If your query does not exist, then you must create it before trying to reference it. For information about creating queries, see *Creating and Saving LFA Queries*.
`SBL-LFA-00114` `Filter for "%s" does not exist.`	The specified parameter cannot be used as a filter.	Do not use this item as a query parameter.
`SBL-LFA-00115` `Category "%s" does not exist.`	You tried to use the specified word, but only key words are expected	Fix the command and try again. For information about key words, see *Listing Query Command Key Words*.
`SBL-LFA-00116` `Object "%s" does not exist.`	The object (that is, Siebel Server, Siebel Application Interface, query, user, component, or session) that you are trying to reference is unavailable.	Make sure the object is available for reference. For information about listing existing objects, see *Listing Log File Analyzer Queries and Run-time Details*.
`SBL-LFA-00117` `Object "%s" already exists. Please use another name.`	An object by that name already exists.	Use another name for your object.
`SBL-LFA-00118` `Query "%s" finished abnormally.`	The query finished abnormally, possibly due to corrupt log files or user intervention.	Re-run the query. If that does not work and the query is complex, then try simplifying it.
`SBL-LFA-00119` `"%s" should not be used for naming.`	The name you have specified cannot be used.	Use another combination of characters.
`SBL-LFA-00120` `Cannot interpret: "%s"`	The name you have specified cannot be used in this place.	The LFA identified an error in your command syntax. For information about valid LFA commands, see *About Running Log File Analyzer Commands*.
`SBL-LFA-00121` `Token has a wrong value: "%s"`	The specified value is invalid.	For information about valid LFA commands, see *About Running Log File Analyzer Commands*.
`SBL-LFA-00122` `Unknown issue.`	There is an error in the command that you have entered.	For information about valid LFA commands, see *About Running Log File Analyzer Commands*.

System or Error Code and Message	Diagnostic Steps or Cause	Solution
`SBL-LFA-00123` `There is no file "%s".`	The input file that you specified when starting the LFA does not exist.	Make sure the file exists and the filename and path is correct.
`SBL-LFA-00124` `Wrong format of the string: "%s".`	The specified string is formatted incorrectly.	For information about valid LFA commands, see *About Running Log File Analyzer Commands*.
`SBL-LFA-00125` `Error parsing configuration file "%s".`	The Log File Analyzer configuration file specified in the message text is missing.	Restart the LFA with another configuration file, or make sure the specified configuration file is available.
`SBL-LFA-00126` `Too many unrelated files are found following main server log file pattern: "%s".`	The log files in the server `log` directory are inconsistent. More than one unrelated file fits the main server log file pattern that is used by the LFA to initialize the server model.	Remove all unrelated files and try again.
`SBL-LFA-00127` `Invalid usage of the command.`	You have used the command incorrectly.	For information and links to the correct usage of LFA commands, see *About Running Log File Analyzer Commands*.
`SBL-LFA-00128` `Component with name "%s" could not be found.`	The Log File Analyzer cannot translate the component name you entered into a component short name.	If this is a valid component, then specify its short name in the LFA configuration file. For more information, see *Configuring the Log File Analyzer*.
`SBL-LFA-00130` `Language "%s" could not be initialized. Please see Log File Analyzer documentation for more information.`	The language files in the `locale` directory on the Siebel Server might be missing or corrupt.	Review information about LFA log file language considerations. For more information, see *About the Log File Analyzer*.
`SBL-LFA-00131` `String with code "%s" could not be loaded. Please see Log File Analyzer documentation for more information.`	The language files in the `locale` directory on the Siebel Server might be missing or corrupt.	Review information about LFA log file language considerations. For more information, see *About the Log File Analyzer*.
`SBL-LFA-00132` `Formatting string "%s" is not supported. Parameters for this string could not be extracted.`	An error in the string makes it impossible for the Log File Analyzer to parse it properly.	If you cannot resolve the underlying issue that caused this error, then contact Oracle Global Customer Support by creating a service request (SR) on My Oracle Support.

7 Collecting Siebel Diagnostic Data

Collecting Siebel Environment Data

This chapter describes how to collect Siebel environment information (such as setup, configuration, and logging information) for diagnostic and troubleshooting purposes using the Siebel Diagnostic Collector (SDDC) command-line utility. This chapter includes the following topics:

About Siebel Diagnostic Data Collector

The Siebel Diagnostic Data Collector (SDDC) is a command-line utility that resides in the binary (`bin`) subdirectory of the Siebel Server, Siebel Gateway, and Siebel Application Interface root directory as the executable siebsnap.exe on Microsoft Windows or as binaries on UNIX. When run, the Siebel Diagnostic Data Collector (SDDC) utility collects information individually for Siebel Servers or the Siebel Gateway. The utility stores the collected data in output files. These files are available for immediate review, or can be sent to Oracle Global Customer Support if required by creating a service request (SR) on My Oracle Support. For information about using SDDC to collect environment data, see *Process of Collecting Siebel Environment Data Using SDDC*.

SDDC creates output files after each execution. These files document environment information for each specific entity. For details on the location and type of collected information, see *About Reviewing Siebel Environment Data*.

> **Note:** To run an environment data collection, make sure you have all necessary executables or binaries available. For more information, see *About SDDC Executables and Binaries*.

About SDDC Executables and Binaries

The Siebel Diagnostic Data Collector (SDDC) utility uses the following executables or binaries to collect environment data comprehensively. The SDDC does not require all executables and binaries to run; however, the SDDC collects the most information when all are present. The executables and binaries are listed for each operating system and platform.

Windows Executables

- odbcsql
- osql (if using MS SQL)
- netstat
- sqlplus (if using Oracle Database)
- db2level (if using db2)

UNIX Binaries (Common)

The SDDC uses the following 31 binaries on all UNIX platforms.

- /usr/bin/cp
- /bin/find
- /bin/hostname
- /usr/bin/ls
- /bin/touch
- /bin/uname
- /bin/tar
- /bin/echo
- /bin/netstat
- /bin/mv
- /bin/sum
- /etc/system
- /bin/compress
- /bin/wc
- /usr/sbin/ndd
- /bin/mkdir
- /bin/head
- /dev/tcp
- /bin/rm
- /bin/coreadm
- db2level
- /bin/chmod
- /bin/sed
- /usr/bin/ipcs
- /bin/grep

- /bin/awk
- db2
- /bin/cat
- /bin/date
- sqlplus
- what

UNIX Binaries for AIX

- lscfg
- instfix
- lsattr
- lsps
- lsfs
- lspv
- lsvg
- no
- ifconfig
- /bin/oslevel
- /bin/getconf
- /bin/lslpp
- /etc/security/limits
- /bin/errpt
- /etc/inittab

UNIX Binaries for Linux

- gcc
- getconf
- /sbin/ifconfig
- /proc/cpuinfo
- /sbin/sysctl

Process of Collecting Siebel Environment Data Using SDDC

You run Siebel Diagnostic Data Collector (SDDC) to collect environment setup, configuration settings, and logging information about the system. For more information about SDDC, see *About Siebel Diagnostic Data Collector*.

Run SDDC separately for the Siebel Servers or the Siebel Gateway to collect information specific to that entity. The kinds of tasks you can perform are similar for both Microsoft Windows and UNIX, but there are additional preparation steps on UNIX before you can begin collecting data.

To collect Siebel environment data using SDDC, perform the following tasks:

1. (UNIX only) Do *Preparing the UNIX Environment to Run SDDC*.
2. Do one of the following:
   - Run SDDC to collect data for the Siebel Server.
     For more information, see *Running SDDC to Collect Siebel Server Data*.
   - Run SDDC to collect data for the Siebel Gateway.
     For more information, see *Running SDDC to Collect Siebel Gateway Data*.

Preparing the UNIX Environment to Run SDDC

Perform the following procedure to prepare the UNIX environment to run Siebel Diagnostic Data Collector (SDDC).

To prepare the UNIX environment to run SDDC

1. Run a database-specific script to set database environment variables.
2. Run the `siebenv.sh` or `siebenv.csh` scripts to set Siebel environment variables.
   For more information about these scripts, see *Siebel Installation Guide* .
3. Change the permissions to execute SDDC.

You are now ready to collect data using SDDC using one of the following procedures:

- *Running SDDC to Collect Siebel Server Data*
- *Running SDDC to Collect Siebel Gateway Data*

Running SDDC to Collect Siebel Server Data

This topic provides procedures for collecting Siebel Server data using Siebel Diagnostic Data Collector (SDDC) on Microsoft Windows and UNIX.

The following table shows additional optional parameters available for use in collecting data on Microsoft Windows. These parameters apply to both Siebel Server and Siebel Gateway.

Parameter	Description
`/c siebsnap.cfg`	Include this parameter to reference a particular configuration file. Use this parameter if Oracle Global Customer Support provides a configuration file. For more information, see *Configuring SDDC Content on Microsoft Windows*.
`/h`	Use this parameter with the siebsnap.exe command to list help information on SDDC and its parameters.
`/l`	Indicates the preferred language.

Parameter	Description

The following table shows additional optional parameters available for use in collecting data on UNIX. These parameters apply to Siebel Server and Siebel Gateway.

Parameter	Description
`-c siebsnap.ini`	Include this parameter to reference a particular configuration INI file. For more details, see *Configuring SDDC Content on UNIX*.
`-help`	Use this parameter with the siebsnap command to list help information on SDDC and its parameters.
`-ouput`	Use this parameter to configure a different SDDC output location. For more details, see *SDDC Output on UNIX*.

Running SDDC to Collect Siebel Server Data on Microsoft Windows

Use the following procedure to collect Siebel Server data on Microsoft Windows.

To run SDDC to collect Siebel Server data on Microsoft Windows

1. Navigate to the binary (`bin`) subdirectory within the Siebel Server root directory.
2. Run `siebsnap.exe` using the `/s` flag, as shown, and, as necessary, additional parameters listed in the table in *Running SDDC to Collect Siebel Server Data*:

   ```
   siebsnap.exe /s
   ```

3. Review the collected information in the `siebsnap` output directory, which is created by the SDDC utility under the `SIEBSRVR_ROOT` directory.

Running SDDC to Collect Siebel Server Data on UNIX

Use the following procedure to collect Siebel Server data on UNIX.

To run SDDC to collect Siebel Server data on UNIX

1. Do *Preparing the UNIX Environment to Run SDDC*.
2. Enter the siebsnap command using the `-s` flag, as shown, and, as necessary, additional parameters listed in *Running SDDC to Collect Siebel Server Data*:

   ```
   siebsnap -s siebel_server_name
   ```

   > **Note:** Alternatively, use `-s this_server`.

3. Review the collected information in the siebsrvr_computer-name_server-name output directory.

Running SDDC to Collect Siebel Gateway Data

This topic provides procedures for collecting Siebel Gateway data using Siebel Diagnostic Data Collector (SDDC) on Microsoft Windows and UNIX.

Running SDDC to Collect Siebel Gateway Data on Microsoft Windows

Use the following procedure to collect Siebel Gateway data on Microsoft Windows.

To run SDDC to collect Siebel Gateway data on Microsoft Windows

1. Navigate to the binary (`bin`) subdirectory within the Siebel Gateway root directory.
2. Run `siebsnap.exe` using the `/g` flag, as shown, and, as necessary, additional parameters listed in the table in *Running SDDC to Collect Siebel Server Data*:

   ```
   siebsnap.exe /g
   ```

3. Review the collected information in the `siebsnap` output directory, which is created by the SDDC utility under the `gtwysrvr` directory.

Running SDDC to Collect Siebel Gateway Data on UNIX

Use the following procedure to collect Siebel Gateway data on UNIX.

To run SDDC to collect Siebel Gateway data on UNIX

1. Do *Preparing the UNIX Environment to Run SDDC*.
2. Enter the `siebsnap` command using the `-g` flag, as shown, and, as necessary, additional parameters listed in the table in *Running SDDC to Collect Siebel Server Data*:

   ```
   siebsnap -g siebel_gateway_name
   ```

   Note: Alternatively, use `-g this_server`.

3. Review the collected information in the *`computer-name`*`_gateway` output directory.

Examples of SDDC Commands

This topic provides examples of Siebel Diagnostic Data Collector (SDDC) commands on Microsoft Windows and UNIX.

Examples of SDDC Commands on Microsoft Windows

Some examples of SDDC commands on Microsoft Windows are:

- `siebsnap.exe /c siebsnapw32.cfg -g`

  This command retrieves Siebel Gateway information using a configuration file named siebsnapw32.cfg.

- **`siebsnap.exe /s`**

  This command retrieves Siebel Server information.

- **`siebsnap.exe /c siebsnapw32.cfg /s`**

  This command retrieves Siebel Server information using a configuration file named siebsnapw32.cfg.

Examples of SDDC Commands on UNIX

Some samples of SDDC commands on UNIX are:

- **`siebsnap -s this_server -u sadmin -p sadmin`**

  This command retrieves Siebel Server information using a username and password.

- **`siebsnap -g gtway1`**

  This command retrieves Siebel Gateway information with a Siebel Gateway name of **`gtway1`**.

About Reviewing Siebel Environment Data

The Siebel Diagnostic Data Collector (SDDC) utility creates output files and directories, as necessary, after each execution of the utility. Manually access these files to review the Siebel environment data, or send the output files to Oracle Global Customer Support for review by creating a service request (SR) on My Oracle Support.

The output files document the environmental setup information, application configurations, and log files if specified. For more information about collecting data using SDDC, see *Process of Collecting Siebel Environment Data Using SDDC*.

About SDDC Output

The SDDC Microsoft Windows utility creates output in the format of a root directory with additional subdirectories and files. For details on SDDC Microsoft Windows output file information and locations, see *SDDC Output on Microsoft Windows*.

The SDDC utility on UNIX creates output in the format of compressed files. For details on SDDC output file information and locations on UNIX, see *SDDC Output on UNIX*.

SDDC uses the following naming convention for the creation of the root directory and filenames:

`ss_{GS|SS}`*`yyyy-mm-dd_hh_mm_ss`*

where:

- ss is siebsnap.
- GS|SS is the Siebel Gateway or Siebel Server.
- yyyy-mm-dd is the year, month, and day.
- hh_mm_ss is the hour, minute, and second based on a 24-hour clock.

For example, the directory or filename `ss_SS2017-05-08_17_10_30` represents information collected for a Siebel Server on 8 May at approximately 5:00 P.M. The directory or filename `ss_GS2017-05-07_14_18_58` represents information collected for the Siebel Gateway on 7 May at approximately 2:00 P.M.

Common SDDC Output Files and Directories

The output from a Siebel Diagnostic Data Collector (SDDC) execution for a Siebel Server or the Siebel Gateway contains common directories and files. The following table provides further descriptions of the information collected in these files and directories.

Files and Subdirectories	Description
ReadMe file	Provides a snapshot of the files copied and directories created during the SDDC execution.
siebsnap log file	Provides a detailed log file of information collected during the SDDC execution. This file is only available for SDDC on Microsoft Windows.
Configuration file	Copies the configuration file used if one is specified during the SDDC execution. This file is only available for SDDC on Microsoft Windows.
siebel_info directory	Directory for Siebel environment information. This directory contains further subdirectories, which contain log files and details on the Siebel environment.
system_info directory	Directory for system information. This directory contains text files containing information on hardware, network statistics, operating system, and registry keys.
db_info directory	Directory for database version information. This directory contains text files containing details on the database version.

SDDC Output on Microsoft Windows

On Microsoft Windows, Siebel Diagnostic Data Collector (SDDC) output consists of files stored within a directory structure created by the utility. If a configuration file is not specified, then the default directory for the SDDC output on Microsoft Windows is the `siebsnap` directory under the Siebel Server root. To configure a different SDDC output location, update the OutputDirectory parameter in the SDDC configuration file. For information about configuring this parameter and other parameters in the SDDC configuration file, see *Configuring SDDC Content on Microsoft Windows*.

SDDC creates additional directories within the `siebsnap` directory (or the configured output directory) based on whether SDDC collects data for a Siebel Server or the Siebel Gateway. For details on the time-sensitive directory naming convention for these root directories, see *About Reviewing Siebel Environment Data*.

For locations of the output contents produced for these entities, see:

- *Siebel Server SDDC Output on Microsoft Windows*
- *Siebel Gateway SDDC Output on Microsoft Windows*

For descriptions of the files and directory content of the SDDC output, some of which are common between each entity, see *Common SDDC Output Files and Directories*.

Siebel Server SDDC Output on Microsoft Windows

With a Siebel Diagnostic Data Collector (SDDC) execution for Siebel Server, the utility creates the root Siebel Server output directory, in the format `ss_SSyyyy-mm-dd_hh_mm_ss`, within the `siebsnap` directory (or configured output directory). Within this directory, the utility creates a directory of the format siebsrvr_server_name, where server_name represents the name of the Siebel Server profiled by the utility. The directory structure and contents appear as follows:

```
ss_SSyyyy-mm-dd_hh_mm_ss\
 siebsrvr_enterprise-name_server-name\
 Readme file
 Siebsnap log file
 Configuration file
 system_info\
 siebel_info\
 db_info\
```

Siebel Gateway SDDC Output on Microsoft Windows

With a Siebel Diagnostic Data Collector (SDDC) execution for Siebel Gateway, the utility creates the root Siebel Gateway output directory in the format `ss_GSyyyy-mm-dd_hh_mm_ss` within the `siebsnap` directory (or configured output directory). Within this directory, the utility creates a directory named `gateway`, which collects information on the Siebel Gateway. The directory structure and contents appear as follows:

```
ss_GSyyyy-mm-dd_hh_mm_ss\
 gateway\
 Readme file
 Siebsnap log file
 Configuration file
 system_info\
 siebel_info\
```

SDDC Output on UNIX

On UNIX, Siebel Diagnostic Data Collector (SDDC) output consists of files compressed within a directory structure created by the utility. The default directory for the compressed files is the directory from which SDDC is run. To configure a different SDDC output location, use the `-o` parameter during the SDDC execution. For details on running the SDDC utility on UNIX, see *Process of Collecting Siebel Environment Data Using SDDC*.

The compressed output files have the extension .tar.z appended to the filename created by SDDC using the SDDC output naming convention. For a description of this naming convention, see *About Reviewing Siebel Environment Data*. The extensions .logarchive.tar.z, asserts.tar.z, and logarchive_asserts.tar.z also apply based on the log parameters specified during execution.

For descriptions of the output for each entity see:

- *Siebel Server SDDC Collector Output on UNIX*
- *Siebel Gateway SDDC Output on UNIX*

For descriptions of the files and directory content of the SDDC output, some of which are common between each entity, see *Common SDDC Output Files and Directories*.

Siebel Server SDDC Collector Output on UNIX

With a Siebel Diagnostic Data Collector (SDDC) execution for Siebel Server, the utility creates the compressed file in the format `ss_SS_yyyy-mm-dd_hh_mm_ss.tar.z` in the default output directory (or configured output directory). The information collected by the SDDC utility varies based on the parameter settings in the siebsnap.ini file. For information about configuring the siebsnap.ini file, see *Configuring SDDC Content on Microsoft Windows*.

By default, the Siebel Server SDDC execution collects system information (system_info), database version information (database_info), and Siebel environment information (Siebel_info). For descriptions of the files and directory content, see *Common SDDC Output Files and Directories*.

Siebel Gateway SDDC Output on UNIX

When Siebel Diagnostic Data Collector (SDDC) executes on a Siebel Gateway, it creates the compressed file in the format `ss_GS_yyyy-mm-dd_hh_mm_ss.tar.z` in the default output directory (or configured output directory). The information collected by the SDDC utility varies based on the parameter settings in the siebsnap.ini file. For information about configuring the siebsnap.ini file, see *Configuring SDDC Content on Microsoft Windows*.

By default, when SDDC executes on a Siebel Gateway it collects system information (system_info) and Siebel environment information (Siebel_info). For descriptions of the files and directory content, see *Common SDDC Output Files and Directories*.

Configuring SDDC Content on Microsoft Windows

On Microsoft Windows, Siebel Diagnostic Data Collector (SDDC) can be configured to modify or enhance the amount of information collected during an SDDC execution. A Microsoft Windows SDDC configuration file is required by SDDC to modify any configurations to the output. The configuration file is referenced during the SDDC execution.

By default, a configuration file is not included with the SDDC utility. It is recommended that you contact Oracle Global Customer Support before using configuration files by creating a service request (SR) on My Oracle Support. Oracle Global Customer Support provides configuration files based on the specific information you require. For information about SDDC configurations on UNIX, see *Configuring SDDC Content on UNIX*.

The SDDC configuration file is divided into sections that can be used to configure the type of information and log files collected by the utility. Edit the configuration file with a text editor. The following table provides the SDDC configuration file parameters.

Section	Parameter	Specifies
[Main]	OutputDirectory	Specifies the directory location for the creation of the SDDC directory and output files.
	CollectLog	Specifies whether log files are collected.
	CollectLogArchive	Specifies whether log archive files are collected.
	CollectCrash	Specifies whether failure files are collected.
	CollectStderrFiles	Specifies whether standard error files are collected.

Section	Parameter	Specifies
	CollectDump	Specifies whether back-up system files are collected.
	CollectAssert	Specifies whether assert and prefer files are collected.
	SiebelBinDir	Specifies the directory location of the **SIEBSRVR_ROOT** binary (**bin**) directory.
[Registry]	Key01	Specifies a registry key for collection.
	Key02	Specifies a registry key for collection.
	Key03	Specifies a registry key for collection.
[CrashFiles]	StartDate	Specifies the start date for a range of process failure files to collect.
	EndDate	Specifies the end date for a range of process failure files to collect.
	MatchingFiles	Specifies the process failure file extensions to collect. You can specify the collection of Siebel Flight Data Recorder (FDR) files in this section by identifying the extension FDR (for example, ***.fdr**).
[StderrFiles]	StartDate	Specifies the start date for a range of standard error files to collect.
	EndDate	Specifies the end date for a range of standard error files to collect.
	MatchingFiles	Specifies the standard error file extensions to collect.
[ProcessDump]	StartDate	Specifies the start date for a range of dump files to collect.
	EndDate	Specifies the end date for a range of dump files to collect.
	MatchingFiles	Specifies the dump file extensions to collect.
[AssertFiles]	StartDate	Specifies the start date for a range of assert files to collect.
	EndDate	Specifies the end date for a range of assert files to collect.
	MatchingFiles	Specifies the assert file extensions to collect.
[LogFiles]	StartDate	Specifies the start date for a range of log files to collect.
	EndDate	Specifies the end date for a range of log files to collect.

Section	Parameter	Specifies
	MatchingFiles	Specifies the log file extensions to collect. You can specify the collection of Siebel Application Response Measurement (Siebel ARM) files in this section by identifying the file extension SARM (for example, `*.sarm`).
[LogArchive]	NumArchives	Specifies that SDDC collects log archive files from the `NumArchives` directory.
	MatchingArchiveDir	Specifies the archive directories for collection.
[SiebelServer]	LogDir	Specifies the Siebel Server log directory in the case of not being able to connect to the Siebel Gateway.
	LogArchiveDir	Specifies the Siebel Server log archive directory in the case of not being able to connect to the Siebel Gateway.
[GatewayServer]	LogDir	Specifies the Siebel Gateway in the case the directory name is different than the default.

About SDDC Parameter Configuration

The StartDate, EndDate, and MatchingFiles parameters, which appear in several Siebel Diagnostic Data Collector (SDDC) configuration file sections, have common configuration details as shown in the following table.

Common Parameters	Configuration Details
StartDate, EndDate	Set these parameters to specify collection of data between the two dates. If StartDate and EndDate are set, then do not set the MaxNumFiles parameter. Configure the dates in the following format: `dd-Month_Acronym-yyyy` where: • dd is the integer of the date ranging from 01 to 31. • Month_Acronym is the three-letter month acronym as follows: Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec. • yyyy is the integer of the year. Another valid configuration selection for the StartDate and EndDate parameters is NONE. If NONE is entered for StartDate and a valid date is entered for EndDate, then files prior to the end date are collected. If NONE is entered for EndDate and a valid date is entered forStartDate, then files from the start date to the current date are collected.
MatchingFiles	Set this parameter to collect multiple file formats using a comma-delimited list. Wildcard characters are also applicable. For example, to collect files containing `siebmtsh` in the filename with the extension .dmp and files of the type siebmtshmw5409.dmp, enter: `MatchingFiles=siebmtsh*.dmp,siebmtshmw5409.dmp`

Example of a Microsoft Windows SDDC Configuration File

The following listing is an example of a Microsoft Windows Siebel Diagnostic Data Collector (SDDC) configuration file. For parameter descriptions and configuration details, see *Configuring SDDC Content on Microsoft Windows*.

```
[Main]
OutputDirectory=D:\siebel\SiebelAI\siebsnap
CollectLog=TRUE
CollectLogArchive=TRUE
CollectCrash=TRUE
CollectStderrFiles=TRUE
CollectDump=TRUE
CollectAssert=TRUE
SiebelBinDir = D:\siebel\SiebelAI\bin

[Registry]
Key01 = HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Tag
Key02 = HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Internet Explorer\Version
Key02 =
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\MaxHashTableSize
Key03 =
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\MaxFreeTcbs
Key04 =
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\MaxUserPort

[CrashFiles]
StartDate=05-May-2017
EndDate=10-May-2017
MatchingFiles = crash*.txt

[StderrFiles]
StartDate=05-May-2017
EndDate=10-Jun-2017
MatchingFiles = stderrout_*.txt

[ProcessDump]
StartDate=05-May-2017
EndDate=10-Dec-2017
MatchingFiles = *.dmp

[AssertFiles]
StartDate=05-Dec-2017
EndDate=10-Dec-2017
MatchingFiles=siebel_prefer*,siebel_assert*

[LogFiles]
StartDate=05-Dec-2017
EndDate=10-Dec-2017
MatchingFiles=*.log

[LogArchiveFiles]
StartDate=05-Dec-2017
EndDate=24-Dec-2017
MatchingFiles=*.log

[SiebelServer]
LogDir=M:\siebel\log
LogArchiveDir=M:\siebel\logarchive

[GatewayServer]
LogDir=M:\siebel\log
```

Configuring SDDC Content on UNIX

You can configure the execution of Siebel Diagnostic Data Collector (SDDC) on UNIX to enhance the amount of information collected during an SDDC execution. Modify the SDDC INI file to record any configuration changes to the SDDC UNIX output.

The SDDC INI file, siebsnap.ini, is in the binary (`bin`) subdirectory of the Siebel Server root directory. To modify this file, open with a UNIX text editor.

For information about SDDC configurations on Microsoft Windows, see *Configuring SDDC Content on Microsoft Windows*.

To configure SDDC to collect enhanced diagnostic information

1. With a text editor, open the siebsnap.ini file located in the binary (`bin`) subdirectory of the Siebel Server root directory.
2. Set specific parameters in the siebsnap.ini file based on how much information you require.

   For details and descriptions of SDDC INI file parameters, see the following table.
3. Save the siebsnap.ini file.

INI File Parameter	Description	Default
OutputDirectory	Set this parameter to send the SDDC output to a different file location than the default.	The directory from which SDDC runs.
CollectLog	Set this parameter to TRUE to collect log file information.	TRUE
CollectLogArchive	Set this parameter to TRUE to collect log archive information.	TRUE
CollectCrash	Set this parameter to TRUE to collect process failure file information.	TRUE
CollectDump	Set this parameter to TRUE to collect dump file information.	TRUE
CollectAssert	Set this parameter to TRUE to collect assert file information. For more information about assert files, see *About Other Siebel Server Log Files*.	TRUE
CollectFDR	Set this parameter to TRUE to collect Flight Data Recorder (FDR) file information. For more information about these log files, see *About Flight Data Recorder Log Files*.	TRUE

INI File Parameter	Description	Default
CollectSARM	Set this parameter to TRUE to collect Siebel Application Response Measurement (Siebel ARM) information. For more information about these Siebel ARM files, see *Siebel Performance Tuning Guide* .	FALSE
CollectQuickFix	Set this parameter to TRUE to collect the following quick fix files if they are present: upgrade.txt, obsolete.txt, incompatible.txt, and log.txt.	TRUE
FileRetention	Set this parameter to the number of .tar.z files that you want to retain. It is useful to retain snapshots of the system in regular intervals and compare them. Once SDDC reaches the value set by the FileRetention parameter, it overwrites the oldest file.	2
StartDate, EndDate	Set these parameters to allow the SDDC utility to collect files between a range of dates. Configure the date values in the following format: **dd-Month_Acronym-yy** where: o dd is the integer of the date ranging from 01 to 31. o Month_Acronym is a three-letter month acronym as follows: Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec. o yy is the integer of the last two digits of the year. If no value is set for EndDate, then all files are collected for the current date.	EndDate =Current Date
StartTime, EndTime	Set these parameters in conjunction with the StartDate and EndDate parameters to further refine the range of files collected by the SDDC utility. Configure the time values in the 24-hour clock format. If no values are set, then the default start time is 00:00 and the default endtime is 23:59.	StartTime =00:00, EndTime =23:59

Example of a UNIX SDDC Configuration INI File

The following listing is an example of the contents of a UNIX Siebel Diagnostic Data Collector (SDDC) configuration INI file. For parameter descriptions and configuration details, see *Configuring SDDC Content on UNIX*.

```
OutputDirectory=
CollectLog=TRUE
CollectLogArchive=TRUE
CollectCrash=TRUE
CollectDump=TRUE
CollectAssert=TRUE
CollectFDR=TRUE
CollectSARM=FALSE
```

```
CollectQuickFix=TRUE
FileRetention=2
StartDate=01-May-17
StartTime=00:00
EndDate=20-May-17
EndTime=12:59
```

8 List of Statistics and State Values

List of Statistics and State Values

This chapter contains listings and brief descriptions of Siebel application statistics and state values. It includes the following topics:

Related Topics

About Siebel Application Statistics

About Siebel Application State Values

List of Siebel Server Infrastructure Statistics

The following table lists the statistics defined for the Siebel Server infrastructure. For background information about Siebel application statistics, see *About Siebel Application Statistics*.

Statistic Name	Alias	Description
Avg. Transfer Time	SCBAvgTransferTime	Average time for transferring connection to component
Component Maxed Out Error	SCBCompMaxeOut	Number of times connection transfer failed because component is busy
Component Unavailable Error	SCBCompOffline	Failed to transfer connection due to Component Unavailable

Statistic Name	Alias	Description
Successful Connections	SCBFwdConn	Connection successfully forwarded
Total Connections	SCBTotalConn	Total number of connection attempts
Total Transfer Time	SCBTotalTransferTime	Total time spent transferring connections to component
FDR Buffer Wraps	FDRWraps	Number of buffer wraps
FDR Buffer Life in seconds	FDRBufferLife	Seconds since buffer was created
FDR Avg time between aging	FDRAgingRate	Avg seconds per buffer wrap
CPU Time	CPUTime	Total CPU time for component tasks (in seconds)
Elapsed Time	ElapsedTime	Total elapsed (running) time for component tasks (in seconds)
Maximum Peak Memory Usage	MaxPeakMemory	Peak Mem used by task. Rolls up differently from MinPeakMemory
Minimum Peak Memory Usage	MinPeakMemory	Peak Mem used by task. Rolls up differently than MaxPeakMemory
Sleep Time	SleepTime	Total amount of sleep time for component tasks (in seconds)
Number of Sleeps	Sleeps	Total number of sleeps for component tasks
Total Tasks	TotalTasks	Total number of tasks completed for server components
Tasks Exceeding Configured Capacity	TskXcdCfgCpt	Number of tasks stated that exceeded configured capacity
Num of DBConn Retries	NumDBConnRtrs	Number of retries due to database connection loss
Num of DLRbk Retries	NumDLRbkRtrs	Number of retries due to deadlock rollbacks
Num of Exhausted Retries	NumExhstRtrs	Number of Times All Retries are Exhausted

List of Siebel Application Object Manager Statistics

The following table describes the statistics specific to the Siebel Application Object Manager. For background information about Siebel application statistics, see *About Siebel Application Statistics*.

In the following table, Siebel Application Object Manager session refers to a session between a client and an Application Object Manager. A session begins when the client connects to the Application Object Manager, and ends when the connection is terminated. A session starts a task on the Application Object Manager. If the Application Object Manager's Multithreaded parameter is set to TRUE, then tasks are implemented as threads.

Note: Disregard the following statistics, which are not Application Object Manager-specific but appear in the component statistics view: Avg SQL Execute Time, Number of SQL Executes, Number of SQL Fetches, and Number of SQL Parses.

Statistics Name	Alias	Description
Average Connect Time	AvgConnTime	Average connect time for Object Manager sessions
Average Reply Size	AvgRepSize	Average size of reply messages (in bytes)
Average Request Size	AvgReqSize	Average size of request messages (in bytes)
Average Requests Per Session	AvgReqs	Average number of requests per Object Manager session
Average Response Time	AvgRespTime	Average Object Manager response time
Average Think Time	AvgThinkTime	Average end-user think time between requests
Total Database Response Time	DBRespTime	Total database response and processing time in milliseconds
Object Manager Errors	Errors	Number of errors encountered during Object Manager session
Reply Messages	RepMsgs	Number of reply messages sent by the server
Total Reply Size	RepSize	Total size (in bytes) of reply messages
Request Messages	ReqMsgs	Number of request message received by the server
Total Request Size	ReqSize	Total size (in bytes) of request messages
Total Response Time	RespTime	Total Object Manager response time (in seconds)
Total Think Time	ThinkTime	Total end-user think time (in seconds)

List of Siebel Database Infrastructure Statistics

The following table describes the statistics specific to the Siebel database infrastructure. For background information about Siebel application statistics, see *About Siebel Application Statistics*.

Statistic Name	Alias	Description
Avg SQL Execute Time	AvgSQLExecTime	Average time for SQL execute operations (in seconds)
Avg SQL Fetch Time	AvgSQLFetchTime	Average time for SQL fetch operations (in seconds)
Avg SQL Parse Time	AvgSQLParseTime	Average time for SQL parse operations (in seconds)
SQL Execute Time	SQLExecTime	Total elapsed time for SQL execute operations (in seconds)
Number of SQL Executes	SQLExecs	Total number of SQL execute operations
SQL Fetch Time	SQLFetchTime	Total elapsed time for SQL fetch operations (in seconds)
Number of SQL Fetches	SQLFetches	Total number of SQL fetch operations
SQL Parse Time	SQLParseTime	Total elapsed time for SQL parse operations (in seconds)
Number of SQL Parses	SQLParses	Total number of SQL parse operations
SQL Statement Prepare Time	OM SQL Prepare Time	Total elapsed time for SQL preparation operations. This statistic is used to optimize performance of SQL statements. Time is calculated as the sum of the connector prepare logic time plus the network time plus the database SQL parse time. **Note:** It is recommended that you seek the assistance of your database administrator when optimizing the performance of SQL statements.

List of Siebel EAI Statistics

The following table describes the statistics specific to Siebel EAI. For background information about Siebel application statistics, see *About Siebel Application Statistics*.

Statistic Name	Alias	Description
Siebel Adapter Total Query Calls	SiebAdptTotQueryCalls	Total number of query calls made to the Siebel Adapter
Siebel Adapter Total Query Size	SiebAdptTotQuerySize	Total cumulative size of output property sets (in KB) for all queries

Statistic Name	Alias	Description
Siebel Adapter Total Sync/ Upsert Calls	SiebAdptTotSyncCalls	Total Number of non-query (synchronize, upsert, update or insert) calls made to Siebel Adapter
Siebel Adapter Total Sync Size	SiebAdptTotSyncSize	Total cumulative size of input property sets (in KB) for all non-query calls (synchronize, upsert, update or insert)
EAI Receiver Total Messages Processed	EAIRcvrMsgsProcessed	Total number of messages processed by the EAI Receiver
Total XML Generator Calls	XMLGenTotCalls	Total number of XML generator calls
Total XML Converter Size of Input Buffer	XMLParseTotSize	Total Cumulative Size of Input Buffer (in KB)
Total XML Converter Size of Output Buffer	XMLGenTotSize	Total Cumulative Size of Output Buffer (in KB)
Total XML Parser Calls	XMLParseTotCalls	Total number of XML Parser Calls

List of Siebel Remote Statistics

The following table describes the statistics specific to Siebel Remote. For background information about Siebel application statistics, see *About Siebel Application Statistics*.

Statistic Name	Alias	Statistics Description
Avg node extracted time	AvgTime	Average time per node extracted (in seconds)
Total nodes extracted	TotNodes	Total number of nodes extracted
Total time processing nodes	TotTime	Total time consumed to extract the latest node (in seconds)
Avg node processing time	AvgTime	Average time per node processed (in milliseconds)
Total nodes processed	TotNodes	Total number of nodes processed
Total time processing nodes	TotTime	Total time consumed to process the current node in the current iteration (in milliseconds)
Monitor Period (in seconds)	MonitorPeriod	Advanced: Time duration for which all monitor data are collected and calculated (in seconds)

Statistic Name	Alias	Statistics Description
Current Operation Processing Rate	OperProcessRate	Advanced: Current operations processed per second
Current Position-Rule Operation Processing Rate	PostnOperProcessRate	Advanced: Current Position-Rule operations processed per second
Current Related Visibility-Event Operation Processing Rate	RelVisOperProcessRate	Advanced: Current Related Visibility-Event operations processed per second.
Current Visibility-Event Operation Processing Rate	VisOperProcessRate	Advanced: Current Visibility-Event operations processed per second
Total Operations Processed	TotOper	Advanced: Total operations processed during the monitored period
Total Vis-Event Operations Processed	TotVisOper	Advanced: Total Vis-Event operations processed during the monitored period
Total RelVisEvent Operations Processed	TotRelVisOper	Advanced: Total related Vis-Event operations processed during the monitored period
Total Postn Related Operations Processed	TotPostnOper	Advanced: Total position rule related operations processed during the monitored period
Average Time for Processing a Node	AvgTimePerNode	Average time for processing one node (in milliseconds)
Total nodes processed	TotNodes	Total number of nodes processed
Total time processing nodes	TotTime	Total time consumed to process the current node in the current iteration (in milliseconds)
Average Number of Rows Downloaded	AvgDownloadRows	Advanced: Average number of downloaded records routed during the monitored period
Total Number of Removed Records	TotRecRemove	Advanced: Total number of removed records routed during the last monitored period.
Average Number of Removed Records	AvgRemoveRows	Advanced: Average number of removed records routed during the monitored period
Total Time for Loading Visdata	TotVisdataLoadTime	Advanced: Total time for loading Visdata during the monitored period (in millisecond).
Average Time for Loading Visdata	AvgVisdataLoadTime	Advanced: Average time for loading Visdata during the monitored period

Statistic Name	Alias	Statistics Description
Total Time for Visdata Load SQL	TotVisdataLoadSqlTime	Advanced: Total time for SQL execution for loading Visdata during the monitored period
Average Time for Visdata Load SQL	AvgVisdataLoadSqlTime	Advanced: Average time for SQL execution for loading Visdata during the monitored period
Total Visibility Check SQL Statements Executed	TotalVisCheckSQLExe	Advanced: Total number of Visibility Check SQLs executed for loading Visibility Data database during the last monitored period
Average Time for Waiting Visdata	AvgVisdataWaitTime	Advanced: Average time for waiting Visdata during the monitored period
Total Time for Waiting Visdata	TotVisdataWaitTime	Advanced: Total time for waiting Visdata during the monitored period (in millisecond).
Average Number of VisCheck Load SQL	AvgVisCheckLoadSql	Advanced: Average number of VisCheck SQLs executed for loading Visdata during the monitored period
Total Records Fetched by Visibility Check	TotRecFetchVisCheck	Advanced: Total number of records fetched by Visibility Checks for loading Visibility Data database during the last monitored period
Average Number of VisCheck Load Rows	AvgVisCheckLoadRow	Advanced: Average number of VisCheck SQL records fetched for loading Visdata during the monitored period
Total Number of Visdata Loading	TotVisdataLoads	Advanced: Total numbers of Visdata loading during the monitored period
Total Number of VisData VisChecks	TotvisdataHit	Advanced: Total number of VisChecks that used VisData during the monitored period
Total Number of Visdata Access	TotVisdataAcc	Advanced: Total numbers of Visdata access during the monitored period
Number of Visibility Data Garbage Collection	NumVisDataGC	Advanced: Total numbers of garbage-collection performed on the Visibility Data database during the last monitored period.
Total Number of Visdata FSGC	TotVisdataFSGC	Advanced: Total Number of Visdata Full Scan Garbage Collection during the monitored period
Total Number of Visdata RKGC	TotVisdataRKGC	Advanced: Total Number of Visdata Random Kill Garbage Collection during the monitored period
Hit Ratio of Visibility Data Cache	HitRatioVisData	Advanced: Hit ratio of the Visibility Data cache during the last monitored period

Statistic Name	Alias	Statistics Description
Reconcile-Operations Routed per Period	ReconcileOperRoute	Advanced: Total number of reconcile-operations routed in the last monitored period
Download-Operations Routed per Period	DownloadOperRoute	Advanced: Total number of Download-operations routed during the last monitored period
Remove-Operations Routed per Period	RemoveOperRoute	Advanced: Total number of Remove-operations routed during the last monitored period
Number of Nodes Routed per Second	NumNodeRoute	Advanced: Number of nodes routed per second during the last monitored period.
Total Number of Opers Processed	TotOpers	Advanced: Total number of operations routed during the monitored period
Monitor Period (in Seconds)	MonitorPeriod	Advanced: Time duration for which all monitor data are collected and calculated (in seconds)
Total Number of Nodes Processed	TotNumNode	Advanced: Total number of nodes routed during the monitored period
Operations Routed per Second	OperRoute	Advanced: Number of operations routed per second during the last monitored period
Total Time for TS I/O	TotTSTime	Advanced: Total time for tall/skinny file I/O during the monitored period
Total Number of TS I/O	TotTSAccess	Advanced: Total number of tall/skinny file I/O during the monitored period
Average I/O Time for Tall-Skinny File	AvgIOTSFile	Advanced: Average I/O time for tall-skinny file during the monitored period (in milliseconds).
Total Time for VisData I/O	TotVisdataTime	Advanced: Total time for visdata I/O during the monitored period (in millisecond).
Total Number of VisData I/O	TotVisdataAccess	Advanced: Total number of Visdata I/O during the monitored period
Average I/O Time for Visibility Data File	AvgIOVisDataFile	Advanced: Average I/O time for Visibility Data file during the monitored period (in millisecond).
Total Time for DX File I/O	TotDXFileTime	Advanced: Total time for DX File I/O during the monitored period

Statistic Name	Alias	Statistics Description
Total Number of DX File I/O	TotDXFileAccess	Advanced: Total number of DX File I/O during the monitored period
Average I/O Time for DX File	AvgIODXFile	Advanced: Average I/O time for DX file during the last monitored period (in millisecond).
Total Number of SQLs	TotNumSQLs	Advanced: Total number of SQLs executed during the monitored period
Average Number of SQLs	AvgNumSqls	Advanced: Average number of SQLs executed per operation routed during the monitored period
Total Time for Visibility Check	TotTimeVisCheck	Advanced: Total time spent for Visibility Check during the last monitored period (in millisecond).
Average Time for Vis-Check	AvgVisCheckTime	Advanced: Average time for Vis-Check per operation routed during the monitored period
Total Time for Reconcile	TotReconcileTime	Advanced: Total time needed for reconcile during the monitored period
Average Time for Reconcile	AvgReconcileTime	Advanced: Average time needed for reconcile during the monitored period
Total Time for Performing Related Visibility Check	TotTimeRelVisCheck	Advanced: Total time spent for performing Related Visibility-Check during the last monitored period (in millisecond).
Average Time for Related Vis-Check	AvgRelVisCheckTime	Advanced: Average time needed for Related Vis-Check during the monitored period
Total Time for Download	TotTimeDownload	Advanced: Total time spent on downloading records the last monitored period (in millisecond).
Average Time for Download	AvgDownloadTime	Advanced: Average time for downloading records during the monitored period
Total Time for Reconcile VisCheck	TotRecVisCheckTime	Advanced: Total time needed for reconcile VisCheck during the monitored period
Average Time for Recocile Vis-Check	AvgRecVisCheckTime	Advanced: Average time needed for reconcile vis-check during the monitored period
Total Number of Records Downloaded	TotRecDownload	Advanced: Total number of downloaded records routed during the last monitored period.

List of Siebel Communications Server Statistics

The following table describes the statistics specific to Siebel Communications Server. For background information about Siebel application statistics, see *About Siebel Application Statistics*.

Statistic Name	Alias	Description
Events Processed	EventsProcessed	Total number of events processed
Events Processed Rate	EventsProcessedRate	Rate of Processing the events

List of Siebel Assignment Manager Statistics

The following table describes the statistics specific to Siebel Assignment Manager. For background information about Siebel application statistics, see *About Siebel Application Statistics*.

Statistic Name	Alias	Description
Number of object rows assigned	Number of rows assigned	This statistic represents the cumulative number of records assigned by this component since the server was started.

List of Siebel Workflow Manager Statistics

The following table describes the statistics specific to Siebel Workflow Manager. For background information about Siebel application statistics, see *About Siebel Application Statistics*.

Statistic Name	Alias	Description
Number Requests	NumRequests	Total Number of requests processed
Policy Violations	Violations	Total Number of policy violations

List of Siebel Server Infrastructure State Values

The following table describes the state values specific to the Siebel Server infrastructure. For background information about Siebel application state values, see *About Siebel Application State Values*.

State Value Name	Alias	Level	Description
Number of notification messages processed	NumNotifyMsgsProcessed	Component	Number of notification messages processed
Number of notification messages received	NumNotifyMsg	Component	Number of notification messages received over the pipe
Number of successful notification handler invocations	NumSuccessHndlrNotifications	Component	Number of successful notification handler invocations
Number of failed notification handler invocations	NumFailedHndlrNotifications	Component	Number of failed notification handler invocations
Component Disable Time	CompDisableTime	Component	Timestamp of when the component was disabled
Component Enable Time	CompEnableTime	Component	Timestamp of when the component was most recently enabled
Component Start Time	CompStartTime	Component	Timestamp of when the component was started
Component Status	CompStatus	Component	Current status of the server component
Component Stop Time	CompStopTime	Component	Timestamp of when the component was shutdown
Component Tasks	CompTasks	Component	Current running tasks for the server component
Task Idle	TaskIdle	Task	TRUE, if task is idle
Task Label	TaskLabel	Task	Identifying label for this task
Task Memory Used	TaskMemory	Task	Current amount of memory used by task

State Value Name	Alias	Level	Description
Task Pause Time	TaskPauseTime	Task	Timestamp of when the task was paused
Task Start Time	TaskStartTime	Task	Timestamp of when the task was started
Task Ping Time	TaskPingTime	Task	Timestamp of when the task was last known to be active
Task Resume Time	TaskResumeTime	Task	Timestamp of when the task was most recently resumed
Task Schedule Time	TaskSchedTime	Task	Timestamp of when the task was scheduled
Task Status	TaskStatus	Task	Current status of the task
Task Stop Time	TaskStopTime	Task	Timestamp of when the task was shutdown
User Name	User	Task	Database user name for the task
Disk Full State	DiskFullState	Component	This state value updates when the disk full state is reached during logging.
SCB Deregistration time	SCBDeregTime	Component	Time of last deregistration
Max. Transfer Time	SCBMaxTransferTime	Task	Maximum time for transferring connection to component
Min. Transfer Time	SCBMinTransferTime	Task	Minimum time for transferring connection to component
Server Non-Essential Tasks	NonEssientialTasks	Server	Total Non-Essential running tasks for the server
Server Disable Time	ServerDisableTime	Server	Timestamp of when the Siebel Server was disabled
Server Enable Time	ServerEnableTime	Server	Timestamp of when the Siebel Server was most recently enabled
Server Start Time	ServerStartTime	Server	Timestamp of when the Siebel Server was started

State Value Name	Alias	Level	Description
Server Status	ServerStatus	Server	Current status of the Siebel Server
Server Stop Time	ServerStopTime	Server	Timestamp of when the Siebel Server was shutdown
Server Cipher Strength	SrvrCipherStrength	Server	Server Encryption key length in bits
Server Tasks	SrvrTasks	Server	Total running tasks for the server
Communication Cipher Strength	ComCipherStrength	Component	Communication Encryption key length in bits

List of Siebel Application Object Manager State Values

The following table describes the state values specific to the Siebel Application Object Manager. For background information about Siebel application state values, see *About Siebel Application State Values*.

State Value Name	Alias	Level	Description
Maximum Reply Size	MaxRepSize	Component	Maximum reply message size
Maximum Request Size	MaxReqSize	Component	Maximum request message size
Maximum Response Time	MaxRespTime	Component	Maximum response time for any Object Manager operation
Applet Name	ObjMgrApplet	Task	Current Applet Name
Business Component	ObjMgrBusComp	Task	Current Business Component
Business Service	ObjMgrBusSvc	Task	Current Business Service
View Name	ObjMgrView	Task	Current View Name
Scripting State	ScriptingState	Task	Current VB/eScript Scripting State
Database Login Id	DbLogin	Task	Database Login ID for the current user

List of Siebel EAI State Values

The following table describes the state values specific to Siebel EAI at the task level. For background information about Siebel application state values, see *About Siebel Application State Values*.

State Value Name	Alias	Description
Number of IDOC messages failed to dispatch	NumIdocMsgsDispatchFail	Total number of IDOC messages failed to dispatch
Number of IDOC messages successfully dispatched	NumIdocMsgsDispatchSucc	Total number of IDOC messages successfully dispatched
Number of IDOC messages received	NumIdocMsgsReceived	Total number of IDOC messages received
Number of IDOC messages sent	NumIdocMsgsSent	Total number of IDOC messages sent
Number of IDOCs failed to dispatch	NumIdocsDispatchFail	Total number of IDOCs failed to dispatch
Number of IDOCs successfully dispatched	NumIdocsDispatchSucc	Total number of IDOCs successfully dispatched
Number of IDOCs ignored	NumIdocsIgnored	Total number of IDOCs ignored
Number of IDOCs read	NumIdocsRead	Total number of IDOCs read
Number of IDOCs received	NumIdocsReceived	Total number of IDOCs received
Number of IDOCs sent	NumIdocsSent	Total number of IDOCs sent

List of Siebel Remote State Values

The following table describes the state values specific to Siebel Remote at the task level. For background information about Siebel application state values, see *About Siebel Application State Values*.

State Value Name	Alias	Description
Current node	CurrNode	Current node being extracted
Current node start time	CurrNodeStart	Start time when current node is extracted

State Value Name	Alias	Description
Max time	MaxTime	Maximum time consumed to extract a node (in seconds)
Min time	MinTime	Minimum time consumed to extract a node (in seconds)
Current file num	CurrFileNum	Current file number to be merged
Current node	CurrNode	Current node being merged
First file num	FirstFileNum	First file number to be merged
Last file num	LastFileNum	Last file number to be merged
Max time	Max Time	Maximum process time for a node (in milliseconds)
Min time	MinTime	Minimum process time for a node (in milliseconds)
Node iteration	NodeIter	The iteration number in which the current node is processed
Node start time	NodeStartime	Start time when current node is processed
Time for Txn to be Merged	TimeTxnMerge	Advanced: Elapsed time for a transaction to be merged in the last monitored period (in seconds)
Monitor Period (in Seconds)	MonitorPeriod	The period of time in which the statistic values are calculated
Low Scan Mark	LowScanMark	The lowest transaction ID to start to process
Time for Txn to be Processed	TimeTxnProcess	Advanced: Elapsed time for a transaction to be processed in the last monitored period (in seconds)
Current node	CurrNode	Advanced: Current node (mobile client or regional node) being routed
Current .dx read file	CurrRFile	Current .dx file being read
Current .dx write file	CurrWFile	Current .dx file being written
Current Transaction Id	CurrTxnId	Advanced: Current Transaction ID being routed
Current Node List	CurrNodeList	Advanced: Current list of nodes being routed
Last Update of Node List	LastUpdNodeList	Advanced: Timestamp of the last update of the node list being used

State Value Name	Alias	Description
Time for Transaction to be Routed	TimeTxnRoute	Advanced: Elapsed time for a transaction to be routed in the last monitored period (in seconds)

List of Siebel Communications Server State Values

The following table describes the state values specific to Siebel Communications Server at the component level. For background information about Oracle's Siebel CRM application state values, see *About Siebel Application State Values*.

State Value Name	Alias	Description
Feedback Counter	FeedbackCount	Number of feedback accumulated
Categorization Engine Initialized	Initialized	Include KB loaded
Last Update Time	LastUpdateTime	Last Time KB was updated
Number of Response Groups Loaded	NumResponseGroupsLoaded	Number of response groups currently loaded
Number of Comm Profiles Loaded	NumComm Profiles Loaded	Number of communication profiles currently loaded as part of the currently loaded response groups
Response Groups Loaded	ResponseGroupsLoaded	Response groups currently loaded
Number of busy work queue threads	NumBusyWorkerThreads	Number of busy work queue threads
Send Counter	SendCount	Number of messages sent