Siebel

System Administration Guide

April 2026

Siebel
System Administration Guide

April 2026

F84300-13

Copyright © 1994, 2025, Oracle and/or its affiliates.

Author: Sukanya Mishra

Contents

Get Help i

1 What's New in This Release 1

What's New in Siebel System Administration Guide, Siebel CRM 26.3 Update 1
What's New in Siebel System Administration Guide, Siebel CRM 25.12 Update 1
What's New in Siebel System Administration Guide, Siebel CRM 25.10 Update 1
What's New in Siebel System Administration Guide, Siebel CRM 25.9 Update 2
What's New in Siebel System Administration Guide, Siebel CRM 25.6 Update 2
What's New in Siebel System Administration Guide, Siebel CRM 25.1 Update 2
What's New in Siebel System Administration Guide, Siebel CRM 24.12 Update 3
What's New in Siebel System Administration Guide, Siebel CRM 24.3 Update 3
What's New in Siebel System Administration Guide, Siebel CRM 24.2 Update 4
What's New in Siebel System Administration Guide, Siebel CRM 23.6 Update 4
What's New in Siebel System Administration Guide, Siebel CRM 23.3 Update 4
What's New in Siebel System Administration Guide, Siebel CRM 22.8 Update 5
What's New in Siebel System Administration Guide, Siebel CRM 21.9 Update 5
What's New in Siebel System Administration Guide, Siebel CRM 21.5 Update 5
What's New in Siebel System Administration Guide, Siebel CRM 21.4 Update 6
What's New in Siebel System Administration Guide, Siebel CRM 21.2 Update 6
What's New in Siebel System Administration Guide, Siebel CRM 20.8 Update 7
What's New in Siebel System Administration Guide, Siebel CRM 20.6 Update 7
What's New in Siebel System Administration Guide, Siebel CRM 20.1 Update 8

2 Siebel Enterprise Server Architecture 9

Siebel Enterprise Server Architecture 9
About the Siebel Environment 9
About the Siebel Gateway 10
About the Siebel Enterprise Server 11
About the Siebel Server 11
About Siebel Server Components 14
About the Siebel Application Interface 17
About the Siebel Application Interface Architecture 17

About the Siebel File System and the File System Manager 22
About the Siebel Management Pack 22

3 Configuring the System Architecture 23

Configuring the System Architecture 23
About Configuring the Siebel Server and Its Components 23
About System Environment Variables 24
Configuring System Environment Variables 26
Configuring Siebel Server Load Balancing 28
Configuring the Session Manager 30
Reinstalling the Siebel Gateway and Siebel Server 31

4 Configuring Siebel Servers 33

Configuring Siebel Servers 33
About the Server Manager GUI 33
About Siebel System Parameters 34
Configuring the Siebel Enterprise Server 39
Configuring the Siebel Server 51
Checking Your Siebel Enterprise and Siebel Server Configurations 55
About Siebel Server Component Definitions 55
Process of Creating a Custom Siebel Server Component 56
Deleting a Siebel Server Component Definition 58
Deactivating a Siebel Server Component Definition 59
Advanced Configuration Tasks 59

5 Administering Server System Services 67

Administering Server System Services 67
Starting and Shutting Down a Siebel CRM Deployment 67
Administering the Siebel Gateway System Service 69
Administering the Siebel Server System Service 74
Renaming Servers for an Existing Siebel CRM Deployment 80

6 Administering Siebel Server Run-Time Operations 83

Administering Siebel Server Run-Time Operations 83
Administering Siebel Servers 83
Administering Siebel Server Component Groups 84
Administering Siebel Server Components 85

Administering Component Jobs 87
Administering Siebel Server Tasks 93

7 Using the Siebel Server Manager Command-Line Interface 97
Using the Siebel Server Manager Command-Line Interface 97
Starting the Siebel Server Manager Command-Line Interface 97
Recommendations for Using the Command-Line Interface 101
Siebel Server Manager Commands 102

8 Siebel Server Infrastructure Administration 129
Siebel Server Infrastructure Administration 129
About Server Request Broker (SRBroker) 129
Configuring Tasks for Server Request Broker 130
About Server Request Processor (SRProc) 130
About Siebel Connection Broker (SCBroker) 131
About Other System Management Components 131
Backing Up and Restoring the Siebel Gateway Registry 133
Administering the Siebel File System 134
Rolling Back the Siebel Runtime Repository to a Prior Version 145

9 Application Object Manager Administration 147
Application Object Manager Administration 147
About the Siebel Application Object Manager 147
Configuring the Application Object Manager Environment 148
About Application Object Manager Parameters 149
Administering the Application Object Manager 152

10 Siebel Server Components and Parameters 155
Siebel Server Components and Parameters 155
Siebel Server Component Groups 155
Siebel Server Components 160
Siebel Enterprise, Server, and Component Parameters 168
Correlating Database Sessions 183

11 Parameters in Siebel Application Configuration Files 185
Parameters in Siebel Application Configuration Files 185

Overview of Parameters in Siebel Application Configuration Files 185
About Siebel Application Configuration Files 186
Editing Siebel Application Configuration Files 187
Configuration Parameters Index 188
Description of Siebel Application Parameters 196
Data Source Parameters 199

12 Siebel CRM Event Publication and Subscription 205

Siebel CRM Event Publication and Subscription 205
Overview of Siebel CRM Event Publication and Subscription 205
Configuring Synchronous Publishing 210
Information Flow in Asynchronous Publishing 212
Summary of Event Publication and Subscription Setup Steps 213
Configuration Files for Event Publication and Subscription 214
OCI Streaming with Apache Kafka - Configurations for Integrating Siebel Event Publication and Subscription 240
Unit Testing Event Publication and Subscription 240
Kafka Consumer Application Scalability Guidelines 242
Avro Serialization in Siebel CRM Kafka Integration 242
Recommended Operational Practices 263
Troubleshooting and Debugging 263

13 Siebel CRM and Coherence Enterprise Caching 269

Siebel CRM and Coherence Enterprise Caching 269
Introduction to Coherence Cache 269
Overview of the Coherence Cache in Siebel CRM 270
About Coherence Cache in Siebel CRM 271
SSH Tunneling 271
Overview of Steps to Set Up Coherence with Siebel CRM 272
Installing and Configuring the Coherence Server 272
Coherence Server Health Check 280
Setting Up Siebel Coherence Cache Client 282
Enabling the Feature on the Siebel Server 289
Using the Cache with eScripts 290
Troubleshooting 290
Restrictions 291

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 Administration Guide, Siebel CRM 26.3 Update

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

Topic	Description
Siebel CRM and Coherence Enterprise Caching	Modified Chapter. This chapter explains enhancements to Siebel Enterprise Cache that enable one-time Client Profile deployment across all Siebel Servers with automatic SSH tunnel setup to Coherence.

What's New in Siebel System Administration Guide, Siebel CRM 25.12 Update

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

Topic	Description
Installing and Configuring the Coherence Server	Modified topic: The sub topic *Maximum Cache Size and Guidance on Sizing of Coherence Node Machines* is updated with new information.
Avro Serialization in Siebel CRM Kafka Integration	New topic. This topic explains how to configure Avro serialization for Siebel CRM Kafka integration to optimize message exchange between the Kafka broker and client.

What's New in Siebel System Administration Guide, Siebel CRM 25.10 Update

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

Topic	Description
Correlating Database Sessions	New topics: This topic describes about Object Manager parameter `EnableDbSessCorrelation`.

What's New in Siebel System Administration Guide, Siebel CRM 25.9 Update

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

Topic	Description
Setting Up Siebel Coherence Cache Client	Modified topics: This topic describes Coherence Client setup in SMC.

What's New in Siebel System Administration Guide, Siebel CRM 25.6 Update

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

Topic	Description
Configurations for SASL Authentication	Modified topics: This enhancement enables Siebel CRM Event Publication Subscription to use SASL/OAUTHBEARER for authentication with a Kafka server. The ability to use the popular OAuth 2.0 framework for this integration further expands the options to set up secure integrations with Apache Kafka from Siebel CRM.
Siebel CRM and Coherence Enterprise Caching	New chapter. Describes the use of Coherence Cache with Siebel CRM.

What's New in Siebel System Administration Guide, Siebel CRM 25.1 Update

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

Topic	Description
OCI Streaming with Apache Kafka - Configurations for Integrating Siebel Event Publication and Subscription	New topic. Describes steps to integrate OCI Streaming with Apache Kafka with Siebel Event Publication and Subscription.

Topic	Description
Using Partitions in Kafka with Siebel CRM Event Publication and Subscription	New topic. Describes how to configure Siebel CRM Event Publication and Subscription to use Kafka partitions to improve scalability and enhance the speed of data exchange between Siebel CRM and other integrated applications through parallelism.
Publishing from Server Script	New topic. Lists simplified examples to demonstrate how to send event data to Apache Kafka using a server script and the impact of the **`SingleRecordAsJSONObject`** property.

What's New in Siebel System Administration Guide, Siebel CRM 24.12 Update

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

Topic	Description
Siebel CRM Event Publication and Subscription	Modified topics: This enhancement simplifies the Kafka integration with a JSON message structure sent from Siebel CRM using Runtime Events (RTE).
Leveraging Kafka Partitions	New topic: Describes how leveraging the Kafka partitions with Siebel CRM Event Publication and Subscription improves scalability, supports higher volumes and enhances the speed of data exchange through parallelism.

What's New in Siebel System Administration Guide, Siebel CRM 24.3 Update

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

Topic	Description
Unzipping Siebel File System Files from the Command Line	New topic. To retrieve orphaned file, use the **`sseunzip80`** utility found in the Siebel Server's **bin** folder.
About using an Encrypted Password File	New topic. One time encryption of the password using the **`EncryptString`** utility.
Configurations for SASL Authentication	Modified topic. Introduces support of SASL/PLAIN authentication.

What's New in Siebel System Administration Guide, Siebel CRM 24.2 Update

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

Topic	Description
Configurations for SASL Authentication	New topic. Introduces support of SASL authentication (SASL/SCRAM-SHA-256 and SASL/SCRAM-SHA-512)

What's New in Siebel System Administration Guide, Siebel CRM 23.6 Update

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

Topic	Description
Siebel CRM Event Publication and Subscription	New topic. The Siebel CRM event publication and subscription is a feature that helps you to send information to Kafka from Siebel and receive information into Siebel.

What's New in Siebel System Administration Guide, Siebel CRM 23.3 Update

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

Topic	Description
About the Siebel Application Interface Architecture	New topic. Describes the detailed architecture of the Siebel Application Interface.

What's New in Siebel System Administration Guide, Siebel CRM 22.8 Update

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

Topic	Description
Renaming Servers for an Existing Siebel CRM Deployment	New topic. This topic describes the steps to take when one or more host names have changed for servers that host Siebel CRM server modules in an existing deployment of Siebel CRM.

What's New in Siebel System Administration Guide, Siebel CRM 21.9 Update

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

Topic	Description
About System Environment Variables	Modified topic. The environment variable RESOLV_MULTI is described in *Siebel Installation Guide* .
Siebel Enterprise, Server, and Component Parameters *Generic Parameters*	Modified topics. Added information about the parameter Oracle Degree of Parallelism (alias OraDegreeOfParallelism). This parameter, which is new as of Siebel CRM 21.3 Update and only for Siebel Web Tools and Siebel Tools when used with an Oracle Database, is described in *Siebel Installation Guide* .

What's New in Siebel System Administration Guide, Siebel CRM 21.5 Update

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

Topic	Description
About System Environment Variables	Modified topic. Added mention of the environment variables USE_NEW_MM and USE_NEW_RM. These variables are described in *Siebel Performance Tuning Guide* .

Topic	Description
Siebel Enterprise, Server, and Component Parameters *Generic Parameters*	Modified topics. Added information about the OM - Preload Links for batch components (alias PreloadLinksForBatchComp) parameter. This advanced parameter is new as of Siebel CRM 20.7 Update.

What's New in Siebel System Administration Guide, Siebel CRM 21.4 Update

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

Topic	Description
Backing Up and Restoring the Siebel Gateway Registry	Modified topic. As of Siebel CRM 21.4 Update, an updated version of Apache ZooKeeper is provided for the Siebel Gateway registry. After installation of Siebel CRM 21.4 Update or later, note the following changes in the Siebel Gateway installation location, **`SIEBEL_ROOT\gtwysrvr`**: • The **`zookeeper`** directory has been renamed as the **`registry`** directory. • The zoo1.cfg file has been renamed as the registry.cfg file. This file, formerly located in the **`zookeeper\conf`** directory, is now located in **`registry\conf`**. • The **`version-2`** directory, formerly located in the **`zookeeper`** directory, is now located in **`registry\conf`**. **Note:** The change to a new version of ZooKeeper does not require any customer backup or restore steps. Do not restore a backup of the Siebel Gateway registry from the prior version of ZooKeeper into the new version.

What's New in Siebel System Administration Guide, Siebel CRM 21.2 Update

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

Topic	Description
Troubleshooting Siebel Native Load Balancing	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)

Topic	Description
Siebel Enterprise, Server, and Component Parameters *Generic Parameters*	Modified topics. Added back the OM - Save Preferences (SavePreferences) parameter, which had been removed in a previous revision. In general, it is recommended to set SavePreferences to False for the EAI Object Manager. The parameter description has been updated.

What's New in Siebel System Administration Guide, Siebel CRM 20.8 Update

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

Topic	Description
Configuration Parameters Index	Modified topic. As of Siebel CRM 20.8 Update, Oracle Database SE2 is now used for the local database, replacing Oracle Database XE. The applicable configuration file section is now [LOCAL_SE] instead of [LOCAL_XE]. The sample database is not supported on Oracle Database SE2, and mentions of the [SAMPLE_XE] section have been removed. For more information, see *Siebel Installation Guide* .

What's New in Siebel System Administration Guide, Siebel CRM 20.6 Update

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

Topic	Description
About Siebel Connection Broker (SCBroker)	Modified topic. Updated the descriptions of the LL and RR values for the ConnForwardAlgorithm parameter.
Rolling Back the Siebel Runtime Repository to a Prior Version	New topic. Added topic to reference information about this feature located in the *Siebel Applications Administration Guide* .

What's New in Siebel System Administration Guide, Siebel CRM 20.1 Update

No new features have been added to this guide for this release. This guide has been updated to reflect only product name changes.

2 Siebel Enterprise Server Architecture

Siebel Enterprise Server Architecture

This chapter provides an overview of the Siebel Enterprise Server architecture for Oracle's Siebel CRM. This chapter includes the following topics:

- *About the Siebel Environment*
- *About the Siebel Gateway*
- *About the Siebel Enterprise Server*
- *About the Siebel Server*
- *About Siebel Server Components*
- *About the Siebel Application Interface*
- *About the Siebel File System and the File System Manager*
- *About the Siebel Management Pack*

About the Siebel Environment

Each Siebel CRM environment includes at least the following entities listed in the following table.

Entity	Comments
Siebel Clients	Client types include Siebel Web Client, Siebel Mobile Web Client, Siebel Developer Web Client, and Siebel Mobile applications. The Siebel Tools client is a special-purpose client for developers.
Siebel Gateway	A module that provides persistent and dynamic storage of configuration information for the Siebel Enterprise, Siebel Server, Siebel Application Interface, and other configurable entities.
Siebel Enterprise Server	The logical grouping of one or more Siebel Servers in your deployment. You must configure the Siebel Enterprise before you can configure the Siebel Server.
Siebel Server	The middle-tier platform that supports both back-end and interactive processes for every Siebel client. Siebel Server includes many different types of components, including Siebel Application Object Managers.
Siebel Application Interface	A module that enables communication between Siebel Web Clients and Siebel Servers.
Siebel Database	The Siebel tables, indexes, and seed data installed on the RDBMS. Database client software is also required.
Siebel File System	One or more directories that store physical files used by Siebel clients and Siebel Enterprise Server.

Entity	Comments

The Siebel Enterprise Server environment represents the middle tier within the three-tiered Siebel CRM applications environment, between the Siebel client and the Siebel database. For more information, see *Siebel Deployment Planning Guide* . For installation information, see *Siebel Installation Guide* .

About the Siebel Gateway

The Siebel Gateway coordinates the Siebel Enterprise Server and Siebel Servers. A single Siebel Gateway can support a single Siebel Enterprise Server. The Siebel Gateway provides the persistent backing of Siebel Enterprise Server configuration information, including:

- Definitions and assignments of component groups and components
- Operational parameters
- Connectivity information

Because this information changes, such as during the configuration of a Siebel Server, it is written to the Siebel Gateway registry. At startup, the Siebel Server obtains its configuration information from the Siebel Gateway registry.

The Siebel Gateway provides the dynamic registry for Siebel Server and component availability information. At startup, a Siebel Server within the Siebel Enterprise Server notifies the Siebel Gateway of its availability and stores its connectivity information, such as network addresses, in the Siebel Gateway's nonpersistent (volatile) store. For information about backing up and restoring the Siebel Gateway registry, see *Backing Up and Restoring the Siebel Gateway Registry*.

Enterprise components (including the Server Manager) query the Siebel Gateway for Siebel Server availability and connectivity information. When a Siebel Server shuts down, this nonpersistent information is cleared from the Siebel Gateway.

In a Windows environment, the Siebel Gateway runs as a Windows service. In a UNIX environment, the Siebel Gateway runs as a daemon process. The system process for the Siebel Gateway is siebsvc.exe on Windows and siebsvc on UNIX. Each running Siebel Server has a corresponding Siebel Gateway system process. For information about running the Siebel Gateway, see *Siebel Enterprise, Server, and Component Parameters*.

The Siebel Gateway also includes an application container. The application container process for the Siebel Gateway is tomcat8.exe on Windows and javaw on UNIX. See also *Starting and Shutting Down a Siebel CRM Deployment*.

> **Note:** Each Siebel Gateway can support only one Siebel Enterprise and one Siebel database. For more information, see *Siebel Installation Guide* and *Siebel Deployment Planning Guide* .

Impact of Failure

When the Siebel Gateway goes down, service to active user connections is not immediately interrupted. All of the Siebel Server components, including Application Object Managers, that are currently running continue to do so. However, no new Siebel Server components can be started or added. Server administration functions become limited.

As of Siebel CRM 18.5 Update, Siebel CRM supports an optional native clustering feature for Siebel Gateway to provide high availability benefits to Siebel CRM customers. For more information, see *Siebel Installation Guide* .

About the Siebel Enterprise Server

The Siebel Enterprise Server is a logical grouping of Siebel Servers that supports a group of users accessing a common Siebel database. The Siebel Enterprise Server can be configured, managed, and monitored as a single logical group, allowing the Siebel administrator to start, stop, monitor, or set parameters for Siebel Servers within a Siebel Enterprise Server. You must configure the Siebel Enterprise before you can configure the Siebel Server.

You can set some Siebel Server parameters at the Siebel Enterprise Server level, and these parameters apply to every Siebel Server and component operating within that Siebel Enterprise Server. Other parameters can be adjusted at the Siebel Server or component level to support fine-tuning your deployment.

Parameter settings are inherited unless overrides are in effect, as follows:

- If a parameter is set at the server level, then the server-specific value overrides the Siebel Enterprise Server parameter setting on that server.
- If a parameter is set at the component level, then the component-specific value overrides the enterprise-level or server-level parameter setting on that component.

Each Siebel Server belonging to a Siebel Enterprise Server must connect to the same Siebel database in the same database server.

The Siebel Enterprise Server itself has no processes and, therefore, cannot have a state.

For more information about configuring the Siebel Enterprise Server, see *Configuring the Siebel Enterprise Server*.

About the Siebel Server

The Siebel Server is the middle-tier platform that supports both back-end and interactive processes for every Siebel client. These processes are components within the Siebel Server architecture and support functions like the following:

- Operation of business logic for Siebel Web clients, as well as connectivity and access to the Siebel database and Siebel File System
- Mobile Web Client synchronization
- Integration with legacy or third-party data
- Automatic assignment of new accounts, opportunities, service requests, and other records
- Workflow management

The Siebel Server supports both multiprocess and multithreaded components, and can operate components in background, batch, and interactive modes. Many of the Siebel Server components can operate on multiple Siebel Servers simultaneously to support an increased number of users or larger batch workloads. For more information about Siebel Server components, see *About Siebel Server Components*.

Note: Make sure that your server hardware and software meet minimum standards. For more information, see the Certifications tab on My Oracle Support.

This topic contains the following information:

- *About the Siebel Server System Service*
- *About Siebel Server Manager*

About the Siebel Server System Service

The Siebel Server runs as a system service that monitors and controls the state of every Siebel Server component operating on that Siebel Server. Each Siebel Server is an instantiation of the Siebel Server system service within the current Siebel Enterprise Server.

The Siebel Server runs as a Windows service in a Windows environment and as a daemon process in a UNIX environment. The system process associated with the Siebel Server is siebsvc.exe on Windows and siebsvc on UNIX. Each running Siebel Server has a corresponding Siebel Server system process. For information about administering the Siebel Server system service, see *Administering the Siebel Server System Service*.

During startup, the Siebel Server system service performs the following sequential steps:

- Retrieves configuration information from the Siebel Gateway. For information about the Siebel Gateway, see *About the Siebel Gateway*.
- Creates a shared memory file located in the `admin` subdirectory of the Siebel Server root directory on Windows and the `sys` subdirectory on UNIX. By default, this file has the following name:

```
Enterprise_Server_Name.Siebel_Server_Name.shm
```

The total shared memory consists of a fixed amount for the Siebel Server itself, a block for each server component running on the server, and a block for each task.

Prior to creating the SHM file, the shared memory for the Siebel application executable programs is built up in the RAM of the computer by using the information retrieved from the Siebel Gateway. This process can use significant amounts of memory. After the creation of the SHM file, the Siebel Server system service releases this memory. The Siebel Server system service deletes this file when it shuts down.

> **Note:** If the Siebel Server system service is improperly shut down, then the SHM file might not be deleted by the Siebel Server system service. In this case, delete (or rename) this file before restarting the Siebel Server system service. If this file is not visible, then it might be a hidden file.

- Siebel Connection Broker (alias SCBroker) server component opens a TCP port to accept inbound Application Object Manager requests. If there are multiple instances of SCBroker on this Siebel Server, then all of the instances listen on the same port.
- Opens TCP ports dynamically for non-Application Object Manager components as necessary, such as Workflow Process Manager.
- Forks single-threaded and multithreaded processes for background mode components enabled on the Siebel Server. The previously created ports are inherited to these processes. For more information about these processes, see *About Server Component Processes (Shells)*.
- When server component processes start, each process updates the shared memory table with component availability and status information. SCBroker and SRBroker use this information for load balancing and routing purposes.
- Archives log files by moving the current `log` directory to the `logarchive` directory.

Note: If the `log` or `logarchive` directory is locked or inaccessible, then a log archive is not created.

About Siebel Server Manager

The Siebel Server Manager is the native management and administration interface for the Siebel Server and Siebel Enterprise Server. The Siebel Server Manager allows you to configure the parameters governing the operation of each component, and determine on which Siebel Servers a given component can operate.

Use the Siebel Server Manager to do the following:

- Start, stop, pause, and resume Siebel Servers, components, and tasks.
- Monitor the status and collect statistics across the Siebel Enterprise Server, Siebel Servers, components, and tasks.
- Manage the configuration of the Siebel Enterprise Server, Siebel Servers, components, and tasks.

You can operate the Server Manager by using one of two interfaces:

- The graphical user interface, or GUI, by using the server administration views in the Siebel application client.

  Use the Server Manager GUI for most administrative duties because it includes greater user interface functionality (including the ability to search for and sort various fields within views) and a more intuitive view into the operation of Siebel Servers than does the command-line interface.
- The command-line interface, or the srvrmgr program.
- Use the command-line interface for batch mode processing, because it can run from batch scripts by invoking script files with administration commands that must be run on a regular basis.

The Server Manager (both the GUI and the command-line interface) connects to the Siebel Gateway, which contains availability and connectivity information for the Siebel Servers within the Siebel Enterprise Server. The Server Manager then connects with each of the Siebel Servers and starts a Server Manager component task.

Note the following behavior:

- If you access the GUI, then Server Manager creates a task on every running Siebel Server.
- If you access the command-line interface without specifying a specific Siebel Server, then Server Manager creates a task on every running Siebel Server.
- If you start the command-line interface while specifying a specific Siebel Server, by using the /s or -s flag, then Server Manager creates a task on that specific Siebel Server alone, and all of the commands are targeted to that Siebel Server at the server level.

On each Siebel Server, the Server Manager task:

- Handles administration commands from the Server Manager
- Executes requested functions
- Returns each operation's results to the Server Manager

Note: Each session of Server Manager creates a separate Server Manager task. Therefore, you create a new Server Manager task each time that you access the server administration screens.

About Siebel Server Components

The various programs that operate on the Siebel Server are implemented as *components*. A component represents only a specific type of program. A component is executed or operated as a *task*, or instantiation of a component, on a specific Siebel Server.

This topic contains the following information:

- *About Server Component Modes*
- *About Server Component Types*
- *About Server Component Groups*
- *About Server Component Processes (Shells)*

About Server Component Modes

Components execute tasks in one of three run modes: background, batch, or interactive:

- **Background mode components.** Background mode components execute tasks to perform background operations for the Siebel Server. After a background mode component task starts, it runs until you explicitly stop the task, or until the Siebel Server itself is shut down.

  You can manually start a background mode component by using the Siebel Server Manager. Components with a Default Tasks parameter set to a value greater than zero might start automatically when the Siebel Server is started. Examples of background mode components include Transaction Router, Replication Agent, and Workflow Monitor Agent.
- **Batch mode components.** You must manually start these components by using the component job process in the Server Manager GUI or the Server Manager command-line interface. Batch mode components end after the task has been performed. Examples of batch mode components include Database Extract and Enterprise Integration Manager.
- **Interactive mode components.** Interactive mode components start tasks automatically in response to client requests. Interactive mode component tasks execute for as long as the client maintains the session, and end when the client disconnects. Examples of interactive mode components include Synchronization Manager and Application Object Managers.

For a list of Siebel Server components and their associated run modes, see *Siebel Server Components*.

About Server Component Types

Siebel Server supports multiple component types. Each type performs a specific function or job. A component type is configured with a set of parameters that determine its behavior to create an entity called a *defined component* (or *component*). Components are defined at the Siebel Enterprise Server level in *component groups*. Component groups are then assigned to one or more Siebel Servers within the Siebel Enterprise Server on which they can execute tasks.

When the Siebel Server is installed and initially configured, predefined components are automatically configured for each component type. These predefined components are then automatically assigned to each Siebel Server within the Siebel Enterprise Server. You can run your entire Siebel CRM deployment by using these predefined components, or

you can modify their definitions and create new defined components to fine-tune your Siebel configuration. For a list of predefined Siebel Server components, see *Siebel Server Components*.

The defined components feature allows you to create multiple defined components for a given component type, simplifying the process of starting various types of tasks that use different parameters, and managing components across multiple Siebel Servers.

For example, you might create one defined component for an Application Object Manager for the Siebel Sales application in Norwegian, and another for an Application Object Manager for the Siebel Service application, also in Norwegian. (Norwegian is an example of an *unshipped language*, as discussed in *Siebel Global Deployment Guide*.) Although these two defined components use the same component type, they service distinct sets of users with different functionality requirements, and are distinct entities that can be individually managed, configured, and administered. Defined components are configured in the Enterprise Component Definitions view of the Server Manager GUI.

Note: For the remainder of this guide, the term *component* refers to both predefined components and defined components that you might create or modify.

About Server Component Groups

Component groups are functional areas that involve logical groupings of Siebel Server components and multiple operating system processes. A component group consists of one or more components, which might be running in one or more operating system processes. Component groups act as:

- The unit of deployment on, or assignment to, a Siebel Server. In general, you include in a Siebel Server the group of components that are deployed on one or more servers.
- A unit for monitoring functionality of the interrelated components within the group (you can get a summary of the operational status at the component group, which is determined by the individual states of the constituent components).
- A unit of control allowing you, in a single step, to make available or unavailable several interrelated components, such as those included in the Siebel Remote or Workflow Management component groups.

Siebel CRM provides several predefined component groups. For a list of the components contained within each component group, see *Siebel Server Component Groups*. For information about creating your own component groups, see *Creating a Custom Siebel Server Component Group*.

About Server Component Processes (Shells)

The Siebel Server runs each component in its own separate process (or shell). These shells provide the interface for a component to communicate with shared memory, and use infrastructure facilities for logging, events, networking, and so on.

A shell performs the following actions when it is forked off:

- Initializes the logging and networking facility.
- Determines which component to run. The component is specified as a DLL (personality DLL), which is run by the Siebel Server either as part of the input parameters or as part of a network message.
- Attaches to shared memory.

The Siebel Server forks off an appropriate shell based on the component mode (interactive, batch, or background) and whether the component is object manager-based, multithreaded, or both. The tables in this topic identify the shell types created in various scenarios for interactive mode, batch mode, and background mode components.

> **Note:** To conserve system resources and minimize the number of processes running on the Siebel Server, disable the component groups that you do not plan to run. If you cannot disable a component group because you require components within the group, then you can set the other components within the group that you do not require to Manual Start mode. For information about disabling component groups, see *Unassigning Component Groups on a Siebel Server*. For information about setting a component to start manually, see *About Starting Siebel Server Components*.

Shell Types for Interactive Mode Components

The following table identifies the shell types created for interactive mode components.

Multithreaded	Object Manager-Based	Shell
False	False	siebsess
True	False	siebmtsh
True	True	siebmtshmw

Shell Types for Batch Mode Components

The following table identifies the shell types created for batch mode components.

Multithreaded	Object Manager-Based	Shell (Created at Bootstrap)	Shell (Created at Run Time)
False	False	siebproc	siebsh
False	True	siebprocmw	siebshmw
True	False	siebmtsh	siebmtsh
True	True	siebmtshmw	siebmtshmw

Shell Types for Background Mode Components

The following table identifies the shell types created for background mode components.

Object Manager-Based	Shell (Created at Boot Time)	Shell (Created at Run Time)
False	siebproc	siebsh

Object Manager-Based	Shell (Created at Boot Time)	Shell (Created at Run Time)
True	siebprocmw	siebshmw

Examples of Shells for Siebel Server Components

The following are examples of shells for Siebel Server components:

- A background component that is not object manager-based is brought up in a siebproc shell. For example, Transaction Processor (alias TxnProc).
- An interactive component that is multithreaded and not object manager-based is brought up in a siebmtsh shell. For example, Server Request Broker (alias SRBroker).
- A multithreaded, object manager-based component is brought up in a siebmtshmw shell. For example, Call Center Object Manager for U.S. English (Call Center Object Manager (ENU), alias SCCObjMgr_enu).

Parameters Controlling the Number of Shells

The following parameters configure shell (process) startup for interactive, batch, and background mode components:

- Maximum MT Servers (alias MaxMTServers)
- Minimum MT Servers (alias MinMTServers)
- Maximum Tasks (alias MaxTasks)
- Default Tasks (alias DfltTasks)

For more information about configuring these parameters, see *Siebel Enterprise, Server, and Component Parameters* and *Application Object Manager Parameters in Server Manager*.

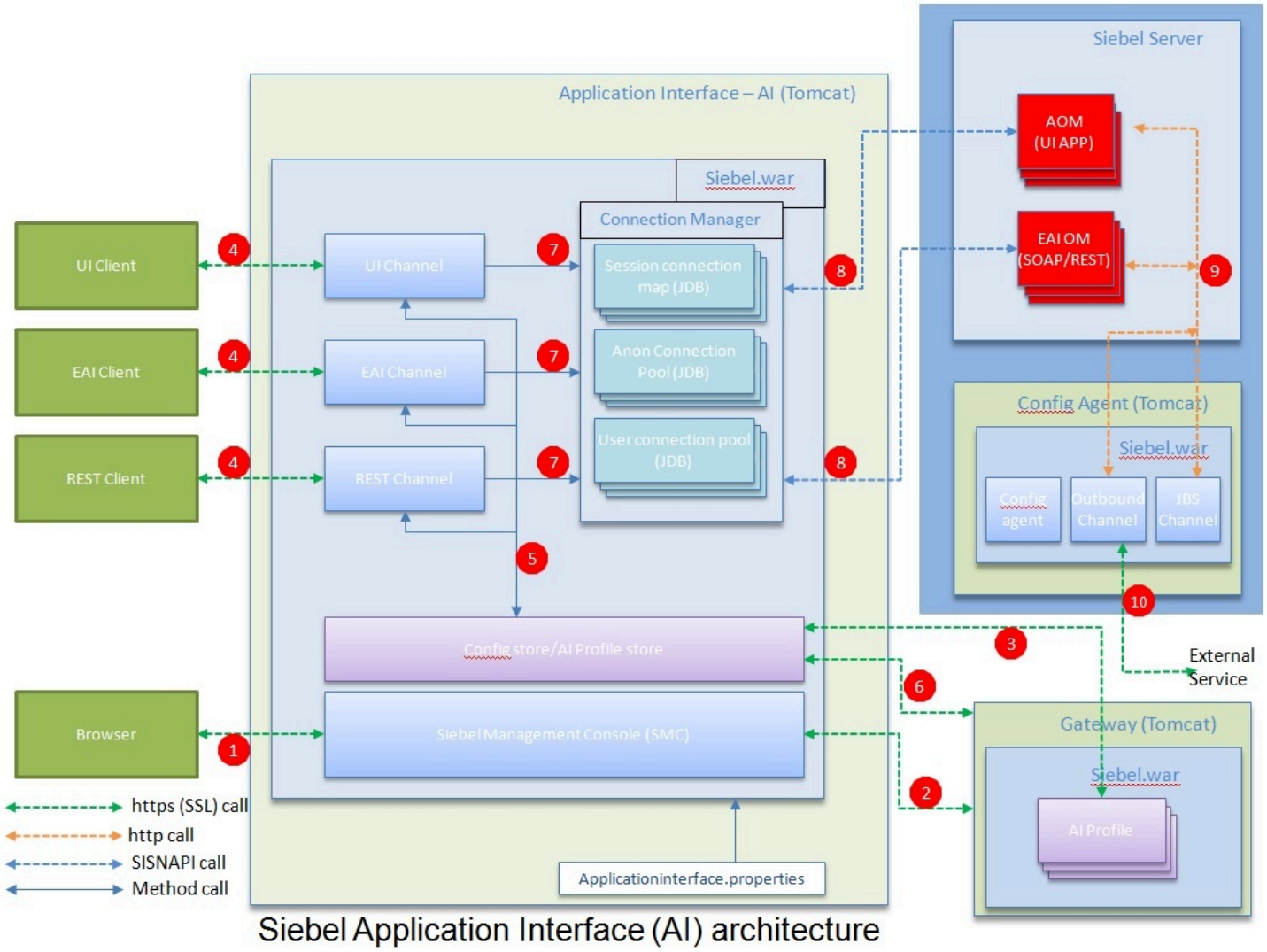
To review information about the shells forked off by the Siebel Server, access the Siebel Server log file. For information about viewing Siebel Server log files, see *Siebel System Monitoring and Diagnostics Guide* .

About the Siebel Application Interface

Siebel Application Interface is a module that enables communication between Siebel Web Clients and Siebel Servers. As of Siebel CRM 17.0, the Siebel Application Interface has replaced the Siebel Web Server Extension (SWSE). For information about installing and configuring Siebel Application Interface, see *Siebel Installation Guide* .

About the Siebel Application Interface Architecture

The following figure shows the detailed architecture of the Siebel Application Interface (AI).



Siebel Application Interface (AI) architecture

The Siebel Application Interface has following major architectural components:

- Siebel Management Console (SMC):
  - This component is a set of UI components that help the user to enter and save the configuration details into the Gateway.
  - Helps to configure and manage various components of Siebel such as Enterprise, Siebel Server, Application interface etc.
  - If an enterprise has multiple AI nodes, then one of them is called Root AI node.
  - The SMC from root AI node is used to configure enterprise, Siebel servers and all AI nodes including itself.
- Config Store or AI Profile Store:
  - This component retrieves and stores the required configuration information from the Gateway.
  - This configuration information is known as an "AI Profile".
  - When AI nodes receives its profile from Gateway (this is called profile push), Config store will store the AI profile name in a properties file known as ApplicationInterface.properties.
  - Whenever AI node is restarted, the config store uses the AI Profile name stored in ApplicationInterface.properties to retrieve the profile from gateway. This called profile pull.
  - Config store also helps in getting the connect string from gateway.
  - This connect string contains the information about the Siebel server to which to connect.
- Application channels:
  - Application channels are the endpoints that are responsible for:
    - Receiving the requests from clients
    - Processing the requests
    - Sending the requests to Siebel server using connections obtained from connection manager
    - Receiving the response from Siebel server
    - Processing the response
    - Sending the response back to clients
  - There are three application channels in AI. They are:
    - UI Channel: Responsible for serving the UI application requests such as callcenter, ecommunication, eservice, epublicsector etc. To these applications generally the client is a browser (such as Chrome, IE or Firefox)
    - EAI Channel: Responsible for serving the EAI related requests such as SOAP web service requests and classic EAI requests. Here the client may be integration handling programs or SOAP UI etc.
    - REST Channel: Responsible for serving the REST requests. Here the clients may be integration handling programs or REST clients such as Postman.
  - Apart from the three channels described above there are two more channels but they are not part of the AI, they are actually are part of config-agent node that runs in every Siebel server node.
    - Outbound Channel
    - JBS Channel
  - All the application channels read their configuration and connect strings from config store.
  - Application channels connect to Siebel server using the connection objects obtained from connection manager.
  - A connection object connects to the server specified in the connect string.

- Connection Manager: Connection manager is responsible for managing the life cycle of connection objects. Each connection object is a JDB object (JAVA Data Bean) which can connect to the Siebel server specified in the connect string.
    - AI has following three types of connections to server:
        - Session based connections: Here a JDB connection to the server will be created for a user when the user logs in and the connection will be kept open until user logs out or session times out. Each connection will be identified by a Session ID. These types of requests are used for UI applications.
        - Anonymous connections: Here a JDB connection to the server will be created using a pre-defined user known as "anonymous user". The anonymous user name and password will be specified while creating the AI profile and connection manager gets the anonymous user name and password from config store. These types of connections are used for EAI anonymous requests. Here connections are identified using sequence numbers.
        - User based connections: Here JDB connections to server are created using the user name and password given in the request. After serving the request the connections are not closed, instead they are retained to be re-used for subsequent requests (may or may not be from same client) that have same user name and password. These types of requests are used for REST requests. Here the connections are identified by user name.

- Connection manager has three components to manage three different types of connections described above.
  - Session connection map: This map stores all the session-based connections to the Siebel server; each connection object is mapped to a session ID.

    When a user accesses an application:
    - The connection manager will request the connect string from config store.
    - Creates JDB connection object which connects to the Siebel server specified in the connect string.
    - Initially this connection logs in to Siebel server using anonymous user name and password obtained from config store.
    - Stores the connection object in the Session connection map with session ID as key.
    - When user gives user name and password, the same connection object will re-login using the user name and password provided by user.
    - This connection object will be used to serve all the requests for that client until user logs off or session expires.
    - Once user logs off or connection expires, the JDB connection object will be removed from the session connection map.
  - Anonymous connection pool: This is a pool of anonymous connections used for EAI SOAP anonymous connections. Whenever an anonymous EAI SOAP request arrives (SOAP request with eai_anon as application and without user name and password):
    - The connection manager looks into this pool for a connection, if this pool has a connection, then it will be used to serve the request. If pool does not have a connection, then a new anonymous connection will be created to serve the request.
    - Once the request is served the connection will be returned to the anonymous pool so that it can be used by other anonymous requests.
    - The number of anonymous connections in the pool will be limited by the anonymous pool size specified in the AI profile.
    - Each anonymous connection will use the anonymous user name and password specified in the AP profile.
  - User connection pool: This is a pool of connections for different users used by REST requests. Whenever a REST request arrives with a valid user name and password:
    - The connection manager will look into the user connection pool for a JDB connection with same user name. If it finds the connection, then the request will be served using this connection and if it doesn't find one, then it will create a new connection to serve the request.
    - Once a request is served, the connection will be returned to the user connection pool so that it can be used for subsequent requests with same user name (may or may not be from same client).
    - Maximum and minimum number of connections in this pool is limited by the Max and Min connections specified in the AP profile.

Siebel Application Interface Functional Steps

When AI is freshly installed and is not configured, it will be in light weight mode. It goes through the following steps in serving the requests. Each of the functional step described below is labelled in red in the above diagram

1. Administrator logs into Siebel Management Console (SMC) using a browser, enters the AI profile parameters, and submits the profile to Gateway.

2. Gateway saves the profile entered by the user.
3. Gateway PUSHES the profile to AI and AI saves the profile in its internal Config store.
4. User sends an application request to AI (the application may be UI or EAI or REST), and the application channel (UI Channel or EAI Channel or REST Channel) will accept the request.
5. Application channel will request the config store to give connect string. Applications get various other parameters as and when needed from config store.
6. Config store sends a request to gateway to provide a connect string (for a given object manager), and gateway will discover the server where the requested object manager is running and provides connect string to AI config store. Config store will return the connect string to application channel.
7. Application Channel requests the connection manager to provide a JDB connection object and provides the connect string and login details. Connection manger provides the JDB connection object that is connected to the Object manager (OM name defined in the connect string) in Siebel server.
8. Channels send the user requests to Siebel server using the JDB connection object and serve the client.
9. If the user request requires execution of a JBS (Java Business Service) or outbound web service, then the object manager will hand over the requests to corresponding channels (JBS or outbound channels) in the config-agent running on the same Siebel sever node.
10. Outbound channel will forward the request to external web service.

About the Siebel File System and the File System Manager

The Siebel File System is a shared directory, or a set of directories on different devices, which is network-accessible to all of the Siebel Servers in the Siebel Enterprise Server. It contains the physical files used by the Siebel clients and Siebel Servers. To gain access to files, Web clients connect to the appropriate Siebel Server to request file uploads or downloads. The Siebel Server then accesses the Siebel File System by using the File System Manager (alias FSMSrvr) component. File System Manager processes these requests through interaction with the Siebel File System directories.

For information about administering the Siebel File System, see *Administering the Siebel File System*. For information about creating the Siebel File System, see *Siebel Installation Guide* .

When using Siebel Developer Web Client for administrative tasks, you might want to connect directly to the Siebel File System without going through the File System Manager.

About the Siebel Management Pack

The Siebel Management Pack provides the underlying infrastructure components that are required to support the deployment features offered by the Oracle Enterprise Manager. The deployment of the Siebel Management Pack is optional. For more information, go to Oracle Technology Network by navigating to the following link about Packaged Application Management solutions. Review all Siebel-related information provided.

```
http://www.oracle.com/technetwork/oem/app-mgmt/index.html
```

In particular, follow the link to *Implementation and Getting Started Guides for all Application Management Suite Products*, and then follow the link to *Getting Started with Application Management Pack for Siebel*.

3 Configuring the System Architecture

Configuring the System Architecture

This chapter provides an overview of configuring the Siebel Server and its components, modifying Siebel Server parameters, and reinstalling the Siebel Gateway and Siebel Servers, if necessary. This chapter includes the following topics:

- *About Configuring the Siebel Server and Its Components*
- *About System Environment Variables*
- *Configuring System Environment Variables*
- *Configuring Siebel Server Load Balancing*
- *Configuring the Session Manager*
- *Reinstalling the Siebel Gateway and Siebel Server*

About Configuring the Siebel Server and Its Components

Before starting the Siebel Server, you might want to modify how it has been configured. You must enable component groups on the Siebel Server. This topic describes some of the configuration-related tasks that you must perform to maintain or administer your Siebel CRM installation.

For more information about configuring Siebel Servers and server components by using the Server Manager GUI, see *Configuring Siebel Servers*. For more information about configuring Siebel Servers and server components by using the Server Manager command-line interface, see *Using the Siebel Server Manager Command-Line Interface*.

> **Note:** The instructions in this chapter assume that you have successfully installed and initially configured the Siebel Gateway, the Siebel Enterprise, and at least one Siebel Server. For more information about performing configuration tasks by using the Siebel Management Console, see *Siebel Installation Guide* .

Before starting the Siebel Server, you might want to add site-specific parameter values or overrides of existing values by using the Server Manager GUI.

You can configure a Siebel Server by modifying the parameters at the Siebel Server, component, or task level for the given Siebel Server. Changes to parameters at the Siebel Server level are inherited at the component and task levels. Changes to parameters at the component level are inherited at the task level. You can also modify parameters at the Siebel Enterprise Server level. Siebel Servers, components, or tasks within the Siebel Enterprise Server inherit the values for modified parameters unless you have explicitly modified (overridden) these parameters at these levels (Siebel Server, component, or task level). For more information, see *About Siebel System Parameters*.

For information about configuration tasks using Siebel Management Console, see *Siebel Installation Guide* .

About System Environment Variables

Environment variables are variables configured for a particular computer hosting an aspect of a Siebel CRM deployment. These Siebel-specific environment variables configure interactions with the computer's operating system and various functions of Siebel CRM applications. Configure environment variables through the operating system of the individual computer. For information about configuring these variables, see *Configuring System Environment Variables*.

The following table provides a partial listing of environment variables for Siebel CRM or that affect the Siebel CRM applications, and it indicates one or more locations where each variable is documented. This list isn't comprehensive. Some environment variables are set automatically. In general, you set environment variables explicitly only as you're instructed by documentation for Siebel CRM.

Environment Variable	Where Documented
JAVA_HOME	*Siebel Installation Guide*
LANG	*Siebel Installation Guide*
LC_ALL	*Siebel Installation Guide*
LD_LIBRARY_PATH (Linux)	*Siebel Installation Guide*
LIBPATH (AIX)	*Siebel Installation Guide*
NLS_DATE_FORMAT	*Siebel Installation Guide*
NLS_LANG	*Siebel Installation Guide*
NLS_SORT	*Siebel Installation Guide*
ODBCINI	*Siebel Installation Guide*
ORACLE_HOME	*Siebel Installation Guide*
PATH	*Siebel Installation Guide*
RESOLV_MULTI (Linux)	*Siebel Installation Guide*
SIEBEL_ASSERT_MODE	*Siebel Performance Tuning Guide*
SIEBEL_CODEPAGE	*Siebel Installation Guide*
SIEBEL_CRASH_HANDLER	*Siebel System Monitoring and Diagnostics Guide*

Environment Variable	Where Documented
SIEBEL_DATA_SOURCE	*Using Siebel Tools*
SIEBEL_DIAG_STORE	*Siebel System Monitoring and Diagnostics Guide*
SIEBEL_ENTERPRISE	*Siebel Installation Guide*
SIEBEL_FILE_ENCODING	*Siebel Marketing User Guide*
SIEBEL_GATEWAY_ROOT	*Siebel Installation Guide*
SIEBEL_HOME	*Siebel Installation Guide*
SIEBEL_LOG_ARCHIVES	*Siebel System Monitoring and Diagnostics Guide*
SIEBEL_LOG_DIR	*Siebel System Monitoring and Diagnostics Guide*
SIEBEL_LOG_EVENTS	*Siebel System Monitoring and Diagnostics Guide*
SIEBEL_OSD_LATCH	*Siebel Performance Tuning Guide*
SIEBEL_OSD_MAXLIMITS	*Siebel Performance Tuning Guide*
SIEBEL_OSD_NLATCH	*Siebel Performance Tuning Guide*
SIEBEL_OSD_PTHREAD_STACK_SIZE	*Siebel Performance Tuning Guide* 2007183.1 (Article ID) on My Oracle Support **Note:** This environment variable applies only to supported versions of UNIX or Linux operating systems.
SIEBEL_ROOT	*Siebel Installation Guide*
SIEBEL_SARMBufferSize	*Siebel Performance Tuning Guide*
SIEBEL_SARMFileSize	*Siebel Performance Tuning Guide*
SIEBEL_SARMLevel	*Siebel Performance Tuning Guide*
SIEBEL_SARMMaxFiles	*Siebel Performance Tuning Guide*
SIEBEL_SARMPeriod	*Siebel Performance Tuning Guide*

Environment Variable	Where Documented
SIEBEL_SERVER_ROOT	*Siebel Installation Guide*
SIEBEL_SESSMGR_TRACE	*Siebel System Monitoring and Diagnostics Guide*
SIEBEL_SISNAPI_TRACE	*Siebel System Monitoring and Diagnostics Guide*
SIEBEL_STDERROUT	*Siebel System Monitoring and Diagnostics Guide*
SIEBEL_TABLE_OWNER	*Using Siebel Tools*
SIEBEL_UNIXUNICODE_DB	*Siebel Installation Guide*
SPELLCHECKLEVEL1	Set to True when using the spell check feature with outbound email messages. For more information about this feature, see *Siebel Email Administration Guide* .
TEMP	*Siebel Installation Guide*
TMP	*Siebel Installation Guide*
TNS_ADMIN	*Siebel Installation Guide*
USE_NEW_MM	*Siebel Performance Tuning Guide*
USE_NEW_RM	*Siebel Performance Tuning Guide*

Configuring System Environment Variables

This topic describes how to configure system environment variables on Windows and on UNIX. For more information about Siebel environment variables, see *About System Environment Variables*.

This topic contains the following information:

- *Configuring an Environment Variable on Windows*
- *Configuring an Environment Variable on UNIX*

Configuring an Environment Variable on Windows

This procedure describes how to configure an environment variable on Windows.

To configure an environment variable on Windows

1. Choose Start, Settings, Control Panel, and then double-click System.
2. Click the Advanced tab, then click Environment Variables.
3. In the System Variables section, click New to create a new environment variable.

   For a partial list of Siebel environment variables, see *About System Environment Variables*.
4. Set the Variable Name field to the name of a Siebel environment variable.
5. Set the Variable Value field to the value for the Siebel environment variable.
6. Restart the computer for the environment variables to take effect.

Configuring an Environment Variable on UNIX

This procedure describes how to configure an environment variable on UNIX.

To configure an environment variable on UNIX

1. Log in as the Siebel Service owner user.
2. Run the siebenv.sh or siebenv.csh script to set Siebel environment variables. For more information about these scripts, see *Siebel Installation Guide* .
3. Depending on the type of UNIX operating system that you use, enter a command like the following to set the environment variable.

   For Korn shell:

   ```
   export Siebel_Environment_Variable=Variable_Value
   ```

   For C shell:

   ```
   setenv Siebel_Environment_Variable
    Variable_Value
   ```

   where:

   - Siebel_Environment_Variable is a Siebel-specific environment variable.
   - Variable_Value is the setting for the environment variable.

   For example, enter the following command for C shell:

   ```
   setenv SIEBEL_SARMLevel 1
   ```
4. Restart the computer for the environment variables to take effect.

Configuring Siebel Server Load Balancing

Siebel Server load balancing distributes the workload across multiple Siebel Servers. For background information and information about initially configuring your Siebel CRM server environment, see *Siebel Deployment Planning Guide* and *Siebel Installation Guide* .

This topic contains the following information:

- *About Rebalancing Siebel Server Loads*
- *Troubleshooting Siebel Native Load Balancing*
- *Verifying Load Balancing Port Access on Siebel Servers*

About Rebalancing Siebel Server Loads

Server loads can become unevenly distributed for several reasons:

- You have just added a new Siebel Server to the network. It will have a low workload compared to other Siebel Servers.
- You have just enabled an Application Object Manager on a Siebel Server. It will have a lower workload than other Application Object Managers on different Siebel Servers.
- There was a server configuration or request routing problem that prevented even distribution of workloads. When this problem is corrected, one or more Siebel Servers will have low workloads.

Siebel native load balancing distributes workloads based on logins. Users must terminate existing sessions and log in to the new sessions to cause workloads to be redistributed. For example, you have 1000 concurrent user sessions running on three Siebel Servers. You then add a fourth Siebel Server. Until all of the users end their sessions and log in again, the load is not evenly distributed between all four servers.

Whenever possible, let normal user login behavior rebalance Siebel Server workloads. Intervene only when absolutely necessary. To rebalance server workloads, stop SCBroker on a Siebel Server. Doing so directs workload away from that server, but does not affect existing user sessions. However, session reconnect does not work for this server. If the connection times out, and user requests come through a Siebel Application Interface other than the one used for login, then the session is lost.

Troubleshooting Siebel Native Load Balancing

This topic provides guidelines for resolving problems with Siebel native load balancing. To resolve a problem, look for it in the list of symptoms or error messages in the following table. Some problem solutions in the table require changing the function of server components.

Problem	Cause	Solution
Users do not get a login page. The browser might display **`Server Busy Error`**.	Verify TCP port access for Siebel Servers, Siebel Gateway, and Siebel Application Interface.	See *Verifying Load Balancing Port Access on Siebel Servers*.

Problem	Cause	Solution
	Increase the Siebel Application Interface logging level.	To turn on detailed Siebel Application Interface logging, increase the log level for Siebel Application Interface. For more information, see the information about configuring the Siebel Application Interface in *Siebel Installation Guide* . The logs for this module are located in `SIEBEL_AI_ROOT/applicationcontainer_external/logs`.
Users can connect but loads are not balanced evenly between Siebel Servers	Unequal loads might be caused by characteristics of users and jobs.	Because jobs are distributed in a round-robin fashion, it is normal for a snapshot of the servers to show somewhat unequal loads. Unequal loads can be caused by several things, including the nature of the jobs and the rate at which users log in and log out on different servers. Over a longer period, the number of sessions handled by each server evens out.
	Siebel Servers do not have equal access to computing resources.	Verify that all of the Siebel Servers have equal access to computing resources such as CPU and memory.
	A Siebel Server has recently been added or has been restarted.	Load balancing is based on user logins. During the process in which current sessions are terminated and new sessions are started, the new Siebel Server is included in the load sharing.
	A Siebel Application Interface cannot route requests to one or more Siebel Servers.	Check for connectivity problems between the Siebel Application Interface and the Siebel Server with the low workload, as described earlier in this table.
	A Siebel Server is rejecting an unusual number of user requests.	Check the Siebel Application Interface log files for `SISNAPI Connection Refused` messages. Possible causes are: • The SCBroker component either is not running or is listening on a different port. • The requested Application Object Manager is not running or cannot run any more tasks. • The requested Application Object Manager has a task or thread that is not responding. • The Application Object Manager cannot communicate with the database server.
	A Siebel Server has functional or configuration problems.	Enable server diagnostics. Look for problems with components. Verify that the basic configuration is correct. For more information about monitoring and diagnosing server problems, see *Siebel System Monitoring and Diagnostics Guide* .

Verifying Load Balancing Port Access on Siebel Servers

This topic describes how to verify access to the load balancing port (that is, the port on which the SCBroker component listens) on your Siebel Servers.

To verify load balancing port access on your Siebel Servers

1. On the computer where Siebel Application Interface is running, telnet to the SCBroker port (such as 2321) on each Siebel Server.

   For example, if a Siebel Server has the host name SiebSrvr1, then use the following command:

   ```
   telnet SiebSrvr1 2321
   ```

   If the connection succeeds, then there is load balancing port access. The connection times out after 500 ms.

   If the connection fails, with the message `Could not open connection to server`, then complete the remaining steps that follow.
2. Verify that the Siebel CRM applications that you want are running on each Siebel Server.
3. On each Siebel Server, verify that SCBroker is running and is configured to listen on port 2321.
4. Verify that the operating system is not blocking access to the SCBroker port.
5. Check that no other networking device, such as a firewall, is blocking access to the SCBroker port.

Related Topics

About Siebel Connection Broker (SCBroker)

Configuring the Session Manager

The Session Manager is a layer within the Siebel Application Interface and Application Object Manager that manages TCP/IP (SISNAPI) connections between clients and Application Object Managers. The Session Manager primarily manages communications between the Siebel Application Interface and the Siebel Server. It is not a stand-alone component, but it is embedded in Siebel Application Interface and in an Object Manager component that is directly accessed through an external interface, such as Component Object Model (COM).

> **Note:** SISNAPI is a proprietary messaging format used for communication with Siebel Servers, and between multiple Siebel Servers. The acronym stands for Siebel Internet Session API (Application Program Interface).

The Session Manager is primarily responsible for establishing and managing connections from the client to the Application Object Manager. A connection is established for each new session request, and is kept open or terminated based on connection multiplexing settings. The Session Manager allows multiple client sessions to share the same pool of connections, thereby minimizing resource usage and maximizing overall performance.

When a client requests a new session, the Siebel Application Interface receives the HTTP request, translates the request into SISNAPI messaging format, and invokes the Session Manager, which is embedded in the Siebel Application Interface, to obtain a physical connection to the Siebel Server. The Session Manager creates a new, temporary connection, which is load-balanced to an available Siebel Server.

After this temporary connection is made to a Siebel Server, Session Manager checks to see whether there are enough connections to the target process. If there are enough TCP connections, then the temporary connection is dropped and the existing connection is used for this new user session. If a connection is not available, then the temporary connection is retained. For more information about this process, see *Siebel Performance Tuning Guide* , which describes the use of the parameter Number of Sessions for each SISNAPI Connection (alias SessPerSisnConn).

After Session Manager opens a connection to the Siebel Server, it closes connections based on the time the connection remains idle. The parameter SISNAPI Connection Maximum Idle Time (alias ConnIdleTime) controls this feature. After reaching the configured idle period, the connection is disconnected by the Application Object Manager process.

A benefit of the ConnIdleTime parameter is to manage connections that pass through a firewall placed between the Siebel Application Interface and the Siebel Server. Because firewalls block idle connections, the ConnIdleTime parameter can be configured to disconnect idle connections before they are blocked by the firewall. This setting avoids future connection problems between the Siebel Application Interface and the Siebel Server.

Note: Only enable the ConnIdleTime parameter at the component level, specifically, for Application Object Manager components.

For more information about communication between the client and the Siebel application, see *Siebel Performance Tuning Guide* .

Parameters that modify Session Manager and SISNAPI connections are available for each component at the component level. For more information about these and other parameters, see *Siebel Enterprise, Server, and Component Parameters*.

Reinstalling the Siebel Gateway and Siebel Server

In some cases, you might have to reinstall the Siebel Gateway and Siebel Servers. These cases include scenarios when you want to:

- Rename the computer on which the Siebel Gateway, Siebel Servers, or both are running.
- Make structural changes to the directory where the Siebel Gateway, Siebel Servers, or both are installed.
- Move the Siebel Gateway, Siebel Servers, or both to another computer.

Each of the listed operations requires you to uninstall and reinstall both the Siebel Gateway and Siebel Servers. Uninstalling also requires that you first perform the necessary tasks to remove associated configuration data. For more information about installing and uninstalling Siebel Servers and the Siebel Gateway and about tasks for configuring and for removing configuration data, see *Siebel Installation Guide* .

CAUTION: In general, uninstalling and reinstalling server modules can present severe consequences for your application environment. Avoid these tasks and associated tasks wherever possible or perform them only when necessary.

You uninstall and reinstall Siebel Servers and the Siebel Gateway in the following sequence.

To uninstall and reinstall server entities

1. Uninstall each Siebel Server.
2. Uninstall the Siebel Gateway.
3. Reinstall the Siebel Gateway.
4. Reinstall each Siebel Server.

   After installation and initial configuration, you perform any additional necessary Siebel Server configuration tasks, such as to define new components, enable or disable components and component groups, assign

component groups to Siebel Server, and so on. You must also reextract all of the Mobile Web Clients. For information about extracting Mobile Web Client databases, see *Siebel Remote and Replication Manager Administration Guide* .

4 Configuring Siebel Servers

Configuring Siebel Servers

This chapter describes Siebel Server configuration tasks and processes that you perform by using the Siebel Server Manager GUI and gives background information about Siebel system parameters. It includes the following topics:

- *About the Server Manager GUI*
- *About Siebel System Parameters*
- *Configuring the Siebel Enterprise Server*
- *Configuring the Siebel Server*
- *Checking Your Siebel Enterprise and Siebel Server Configurations*
- *About Siebel Server Component Definitions*
- *Process of Creating a Custom Siebel Server Component*
- *Deleting a Siebel Server Component Definition*
- *Deactivating a Siebel Server Component Definition*
- *Advanced Configuration Tasks*

About the Server Manager GUI

The Siebel Server Manager graphical user interface (GUI) consists of the views in the following server administration screens, which are available from the application Site Map:

- Administration - Server Configuration
- Administration - Server Management
- Server Jobs

Note: The Siebel Server Manager GUI screens allow you to use only the query operator LIKE.

In general, use the Siebel Server Manager GUI to perform most administrative tasks, because it provides a more intuitive view into the operation of Siebel Servers than does the command-line interface.

Note: By default, the Siebel Server Manager GUI is available on every Siebel client, and is accessible to users with Siebel administrator responsibility. This feature allows Siebel administrators to perform administration tasks from any client on the network. Therefore, it is important to grant the Siebel administrator responsibility only to designated Siebel administrators.

As of Siebel CRM 18.7 Update, Siebel system administrators can configure server elements in the Siebel CRM deployment by using the Configuration screen in the Siebel Management Console. Administrators can perform tasks here that are equivalent to some of the tasks that you traditionally perform in the Administration - Server Configuration

screen in the Siebel application or using the Server Manager command-line utility. The new functionality is provided as an alternative to the traditional methods. For more information, see *Siebel Installation Guide* .

This topic contains the following information:

- *About the Enterprise Explorer View*
- *About the Process Failure Diagnostics View*

About the Enterprise Explorer View

The Administration - Server Configuration screen contains a view for reviewing enterprise data in an Explorer or hierarchical format. The Enterprise Explorer view provides an alternate means of navigation and a comprehensive layout of the enterprise data. You access this view by selecting Enterprise Explorer from the link bar.

About the Process Failure Diagnostics View

The Administration - Server Management screen contains the Process Failure Diagnostics view, which you can use for investigating process failures for Siebel Server components. You access this view by selecting Diagnostics from the link bar.

Related Topics

Using the Siebel Server Manager Command-Line Interface

Related Books

Siebel Installation Guide

Siebel Applications Administration Guide

Siebel Security Guide

Siebel System Monitoring and Diagnostics Guide

About Siebel System Parameters

The Siebel application uses parameter values based on the level at which they are set. Parameter values at the highest levels are inherited by the same parameter at lower levels. For example, a parameter set at the enterprise level contains the same value for the same parameter at the server and component level, unless the value is overridden at a lower level. If a change is made to that parameter at the enterprise level, then this value is inherited down to the lower levels. The table in this topic lists the parameter-setting levels in order from highest to lowest.

If a parameter value is set at a lower level, and a new change is made to the same parameter at a higher level, then the new change does not inherit down to the lower level unless the override is deleted at that lower level.

> **Note:** Do not set lower-level parameters to a blank or empty value. To negate a parameter value, use the appropriate `delete parameter override` command.

Once you set a parameter at a lower level, this value creates an entry in the Siebel Gateway registry and, from that time on, you must maintain it at this level. That is, any further changes that are to affect this level must be made at this level unless you delete the override.

Note: Querying for a specific parameter in either the Parameter field or the Alias field returns matches from both fields. For example, querying in the Parameter field by using the expression File* returns the result Siebel File System, because the parameter's alias is FileSystem.

This topic contains the following information:

- *Hierarchy of System Parameters*
- *About Advanced and Hidden Parameters*
- *About Parameter Availability Status*
- *About Siebel Enterprise Server Parameters*
- *About Siebel Server Parameters*
- *About Siebel Component Parameters*
- *About Task Parameters*
- *About Named Subsystem Parameters*

Related Topics

Deleting System Parameter Overrides

Parameter Management Commands

Hierarchy of System Parameters

The following table lists the parameter-setting levels in order, from highest to lowest. Named subsystem parameters can apply to entities at different levels.

Level	System Parameter	Comment
1	Default from library	Default, hard-coded values from the library. Does not apply for passwords and other parameters that require user-supplied values.
2	Siebel Enterprise parameter	For more information, see *About Siebel Enterprise Server Parameters* and *Configuring Siebel Enterprise Parameters*.
3	Siebel Server parameter	For more information, see *About Siebel Server Parameters* and *Configuring Siebel Server Parameters*.
4	Siebel Server component parameter, enterprise level	Enterprise-level Siebel Server component parameters are set by configuring component definition parameters. For more information, see *Configuring Siebel Enterprise Component Definition Parameters* and *About Siebel Server Component Definitions*.

Level	System Parameter	Comment
5	Siebel Server component parameter, server level	For more information, see *About Siebel Component Parameters* and *Configuring Siebel Server Component Parameters*.
6	Siebel Server component task parameter	Configure these parameters mainly for batch tasks or when invoking tasks from a script or a workflow process. For more information, see *About Task Parameters*, *Starting a Component Job*, and *Configuring Siebel Server Task Dynamic Parameters*.

About Advanced and Hidden Parameters

Parameters that affect or modify advanced product functionality are, by default, hidden from the Server Manager GUI.

To make advanced or hidden parameters visible, click Advanced or Hidden on the parameter views for the enterprise, Siebel Server, or components. To restore the default view, click Reset.

Related Topics

Configuring Siebel Server Component Parameters

List Commands

About Parameter Availability Status

The Server Manager GUI provides parameter availability status for system parameters at the following levels: enterprise, Siebel Server, component definition, component, task, and session. Certain parameters become available or effective only after specific Siebel Server operations are performed. Review the parameter availability status options to determine the type of action necessary to make your parameter change effective.

Availability Status	Description
Immediately	These parameters are effective immediately and require no further Siebel Server operations. These parameter are also known as *dynamic parameters*. All other parameters are *static parameters*.
At Next Task	These parameters are effective at the start of the next task.
At Component Restart	These parameters require a restart of the server component before they become effective.
At Server Restart	These parameters require a restart of the Siebel Server before they become effective.
Require Reconfiguration	These parameters require the component definition to be reconfigured before they become effective.

Related Topics

Reconfiguring Siebel Server Component Definitions

Administering Siebel Servers

Administering Siebel Server Components

Administering Component Jobs

About Siebel Enterprise Server Parameters

Enterprise parameters set the attributes of the entire Siebel Enterprise Server. These parameters are initially set when the Siebel Enterprise Server is configured by using the Siebel Management Console. Each Siebel Server installed and configured in the Siebel Enterprise Server inherits these enterprise parameters. Many of the parameters that can be set at the enterprise level are server or named subsystem parameters, which can then be modified or overridden on each Siebel Server, as needed.

For example, if a Siebel environment contains multiple Siebel Servers, and the component parameters Maximum MT Servers (alias MaxMTServers) and Maximum Tasks (alias MaxTasks) are set at the enterprise level for a specific component with the values 5 and 100, respectively, then, unless override values are in effect, *each* Siebel Server in the environment runs this specific component with a maximum of 5 server processes (MaxMTServers) and a maximum 100 tasks (MaxTasks). The values 100 and 5 do not apply to the enterprise as a whole, but provide values for each instance of the component on an individual Siebel Server.

> **Note:** Setting parameters at the enterprise level sets generic parameters used by all of the Siebel Servers and components across the enterprise. To set component-specific parameters for all components across an enterprise, configure the component definition parameters.

Related Topics

About Siebel System Parameters

About Siebel Server Parameters

Siebel Server parameters set the attributes of each Siebel Server. These parameters are either used by the Siebel Server for its own operation, such as Shutdown Wait Time, or inherited by the components assigned to that Siebel Server.

The entire set of parameters for a given Siebel Server is a combination of the enterprise parameters inherited from the enterprise, and those specified when the Siebel Server is initially configured. Either type can be modified for any given Siebel Server.

If Siebel Server-level parameters are changed, then any future configurations to Siebel Enterprise Server-level parameters do not cascade down to the Siebel Server parameter level for that particular parameter. To restore this functionality, see the description of the appropriate `delete parameter override` command.

New values for Siebel Server-level dynamic parameters, which are parameters marked as Effective Immediately, apply to subsequently started tasks, unless these values are overridden at a lower level.

New values for static parameters, which are parameters not marked Effective Immediately, do not apply to subsequently started tasks until you stop and restart the Siebel Server system service. For both fixed and static parameters, the Server Manager views continue to show both the current value and the value upon Siebel Server restart.

Related Topics

About Siebel System Parameters

Parameter Management Commands

About Siebel Component Parameters

Siebel component parameters set the attributes specific to a particular component type. These parameters are set initially when the defined component is created. For each component assigned to a Siebel Server, the component inherits the Siebel Enterprise and Siebel Server parameters applicable to that Siebel Server. The three types of parameters (except those marked Fixed when the defined component was created) can be overridden for the particular component on that Siebel Server.

If component-level parameters are changed, then any future configurations to Siebel Enterprise Server or Siebel Server-level parameters do not cascade down to the component parameter level for that particular parameter. To restore this functionality, see the description of the appropriate `delete parameter override` command.

New values for component-level dynamic parameters, which are parameters marked as Effective Immediately, apply to subsequently started tasks, unless these values are overridden at a lower level.

New values for static parameters (that is, parameters that are not marked Effective Immediately) do not apply to subsequently started tasks until you stop and restart the Siebel Server system service. For both fixed and static parameters, the Siebel Server Manager views continue to show both the current value and the value upon Siebel Server restart.

Related Topics

About Siebel System Parameters

Parameter Management Commands

About Task Parameters

Task parameters control the execution of a specific task. These parameters consist of Siebel Enterprise, Siebel Server, and component-level parameters for the Siebel Server and the component for which the task is being executed, as well as task-specific parameters specified when you start a task. Task parameters are set or overridden when you first start the task. After a task is running, only dynamic parameters can be changed.

> **Note:** The delay before the new parameter value is picked up and used by a running task varies by component, depending on how often the tasks for a particular component recheck their parameter values.

Related Topics

About Siebel System Parameters

About Named Subsystem Parameters

Named subsystems are groupings of defined enterprise parameters, which allow the Siebel Server to manage multiple sets of parameter values. Like other server constructs, such as component definitions, server parameters, enterprise parameters, and component parameters, they are stored in the Siebel Gateway registry. When a Siebel Server starts, it retrieves this information and creates a copy of the named subsystems in shared memory. You can create named subsystems by using the Server Manager GUI or command-line interface.

The Server Manager GUI also refers to named subsytems as enterprise profiles. You create new named subsystems in the Profile Configuration subview of the Enterprises view in the Administration - Server Configuration screen.

By using named subsystems, the Application Object Manager can maintain several different values for a particular parameter. The value used by the Application Object Manager depends on the context. In other words, an Application Object Manager has several groups of parameters with context-dependent values: in context 1, parameters PA and PB have values V1A and V1B, respectively, whereas in context 2, the same parameters have values V2A and V2B.

For example, the Application Object Manager uses different configuration information that depends on the data source on which the business components are based. Which data source, and data source configuration, is used for a particular business component is context information that can come from several different sources. A business component can specify a data source in the compiled repository file, or a client can select a data source from several available data sources. Configuration information like database case sensitivity can have different values depending on the data source.

The parameters that have a context dependency are defined as named subsystem parameters. The component code that uses these named subsystems can request the subsystem parameter values by using a context name and will receive the value belonging to the named subsystem.

Named subsystem parameters are set at the enterprise level only. Parameter names associated with a data source usually start with DS so they do not conflict with the other parameters from the levels 2 to 6 in the table in *About Siebel System Parameters*. Named subsystem parameters have a higher priority than the default parameter settings that are hard-coded in their library.

Where they apply, named subsystem parameters override parameters set at the Siebel Enterprise, Siebel Server, and server component levels.

Related Topics

About Siebel System Parameters

Creating Siebel Enterprise Server Named Subsystems

Configuring Siebel Enterprise Server Named Subsystem Parameters

Named Subsystem Management Commands

Configuring the Siebel Enterprise Server

This topic lists the configuration tasks applicable to the Siebel Enterprise Server. For more information about the Siebel Enterprise Server and the overall system architecture, see *Siebel Enterprise Server Architecture*. You initially configure the Siebel Enterprise Server modules using Siebel Management Console, as described in *Siebel Installation Guide* .

This topic contains the following information:

- *About Assigned and Unassigned Component Group*
- *Unassigning Component Groups on a Siebel Server*
- *Assigning Component Groups on a Siebel Server*
- *About Enabled and Disabled Component Groups*
- *Enabling Component Groups on a Siebel Enterprise Server*
- *Disabling Component Groups on a Siebel Enterprise Server*
- *Configuring Siebel Enterprise Server Named Subsystem Parameters*
- *Creating Siebel Enterprise Server Named Subsystems*
- *Configuring Siebel Enterprise Parameters*
- *Configuring Siebel Enterprise Component Definition Parameters*
- *About System Alert Notification*
- *Configuring System Alert Notification*
- *Troubleshooting System Alert Notification*
- *Configuring Component Job Templates*
- *Synchronizing Components on a Siebel Enterprise Server*

About Assigned and Unassigned Component Groups

Component groups are assigned to Siebel Servers within a Siebel Enterprise Server. Both predefined and defined components groups are automatically assigned to each Siebel Server installed and configured within an existing Siebel Enterprise Server. Component groups must be assigned to Siebel Servers before tasks can be started for the components belonging to the component group. Only make changes to the component group assignment if you want to unassign or reassign component groups to different Siebel Servers.

Unassigning a component group on a Siebel Enterprise Server results in:

- No allocation of space in the shared memory segment for component groups after startup.
- The removal of the component group entries from the Siebel Gateway data.
- A loss of any component group customization (for example, parameter overrides at the component level).

Changes to the component group assignment state take effect only when the Siebel Server system service and Siebel Gateway system service are restarted.

Generally, only unassign a component group if the component group is not planned for future deployment on a particular server. Alternatively, you can disable a component group temporarily.

> **Note:** Unassign or disable component groups that are not intended to operate on that Siebel Server to reduce unnecessary consumption of server resources. Also, do not initially enable any component groups that are not specifically intended for a given Siebel Server.

Related Topics

Unassigning Component Groups on a Siebel Server

Assigning Component Groups on a Siebel Server

About Enabled and Disabled Component Groups

Disabling Component Groups on a Siebel Enterprise Server

Unassigning Component Groups on a Siebel Server

This topic describes how to unassign component groups on a Siebel Server.

Note: Unassigning a component group from a Siebel Server results in a loss of component group customization, for example, parameter settings.

To unassign a component group on a Siebel Server

1. Shut down the Siebel Server that contains the component group that you want to unassign.
2. Navigate to the Administration - Server Configuration screen, then the Enterprises view.
3. In the Component Groups list, select the component group of interest.
4. In the Component Group Assignments list, select the Siebel Server of interest.
5. Click Unassign.

   The Assigned? field for the specified component group on the specified Siebel Server no longer contains a check mark.
6. Restart the Siebel Server that previously contained the unassigned component group.
7. For the change to take effect, stop and restart the Siebel Server system service and Siebel Gateway system service.

Related Topics

Configuring the Siebel Enterprise Server

About Assigned and Unassigned Component Group

Administering the Siebel Gateway System Service

Administering the Siebel Server System Service

Starting a Siebel Server

Shutting Down a Siebel Server

Assigning Component Groups on a Siebel Server

This topic describes how to assign component groups on a Siebel Server. You can assign component groups that are currently unassigned.

To assign a component group on a Siebel Server

1. Navigate to the Administration - Server Configuration screen, then the Enterprises view.
2. In the Component Groups list, select the component group of interest.
3. In the Component Group Assignments list, select the Siebel Server of interest.

4. Click Assign.

   The Assigned? field for the specified component group on the specified Siebel Server now contains a check mark.
5. For the change to take effect, stop and restart the Siebel Server system service and the Siebel Gateway system service.

Related Topics

Configuring the Siebel Enterprise Server

About Assigned and Unassigned Component Group

Administering the Siebel Gateway System Service

Administering the Siebel Server System Service

About Enabled and Disabled Component Groups

An enabled component group, at the enterprise level and server level, is one of the necessary conditions for execution of server component tasks belonging to that component group. Component groups are enabled and disabled independently at the enterprise and server level.

To enable or disable a component group at the enterprise level, the component group must be assigned on one or more Siebel Servers. To enable or disable a component group at the Siebel Server level, the component group must be assigned on this Siebel Server.

When creating a new component group, first assign the component group to the appropriate Siebel Servers before enabling the component group at the enterprise and server levels.

- **Enabled.** The component group is enabled at the enterprise level. You can then configure the component group run state so tasks can be started for components within the component group.
- **Disabled.** The component group is disabled at the enterprise level. You cannot configure the component group run state, and tasks cannot be started for components within the component group.

Disabling a component group results in:

- Components that are unavailable on Siebel Servers, therefore, tasks cannot be started (existing tasks run to completion).
- No allocation of space in the shared memory segment for those components when the Siebel Server is restarted.

> **Note:** Unassigning or disabling component groups that you do not need will reduce the unnecessary consumption of server resources. If a component group is not planned for immediate deployment on a particular Siebel Server but might be deployed in the future, then you can disable the component group on that Siebel Server. (For a Siebel Server that has not yet been configured, do not enable the component group in the Siebel Management Console.) If a component group is not planned for deployment, then you can remove it from a Siebel Server by unassigning the component group.

For procedures for enabling and disabling component groups, see the following:

- *Enabling Component Groups on a Siebel Enterprise Server*
- *Disabling Component Groups on a Siebel Enterprise Server*

- *About Assigned and Unassigned Component Group*
- *Enabling Component Groups on a Siebel Server*
- *Disabling Component Groups on a Siebel Server*

Related Topics

About Assigned and Unassigned Component Group

Checking Your Siebel Enterprise and Siebel Server Configurations

Enabling Component Groups on a Siebel Enterprise Server

This topic describes how to enable component groups on a Siebel Enterprise Server.

To enable a component group on a Siebel Enterprise Server

1. Navigate to the Administration - Server Configuration screen, then the Enterprises view.
2. In the Component Groups list, select the disabled component group of interest.
3. Click Enable.

   The Enable State field of the component group record changes to Enabled.
4. If the component group contains batch mode components, then synchronize Siebel Server components.
5. For the change to take effect, stop and restart the Siebel Server system service.

Related Topics

Configuring the Siebel Enterprise Server

About Enabled and Disabled Component Groups

Synchronizing Components on a Siebel Enterprise Server

Enabling Component Groups on a Siebel Server

Administering Server System Services

Disabling Component Groups on a Siebel Enterprise Server

This topic describes how to disable component groups on a Siebel Enterprise Server.

To disable a component group on a Siebel Enterprise Server

1. Navigate to the Administration - Server Configuration screen, then the Enterprises view.
2. In the Component Groups list, select the enabled component group of interest.
3. Click Disable.

   The Enable State field of the component group record changes to Disabled.
4. If the component group contains batch mode components, then synchronize Siebel Server components.
5. For the change to take effect, stop and restart the Siebel Server system service.

Related Topics

Configuring the Siebel Enterprise Server

About Enabled and Disabled Component Groups

Synchronizing Components on a Siebel Enterprise Server

Disabling Component Groups on a Siebel Server

Administering Server System Services

Configuring Siebel Enterprise Server Named Subsystem Parameters

This topic describes how to configure Siebel Enterprise Server named subsystem parameters.

To configure named subsystem parameters on a Siebel Enterprise Server

1. Navigate to the Administration - Server Configuration screen, then the Enterprises view.
2. Select the Profile Configuration view tab.
3. In the Profile Configuration list, select the named subsystem (profile) of interest.
4. In the Profile Parameters list, select the parameter of interest and configure its value.

Related Topics

About Named Subsystem Parameters

Configuring the Siebel Enterprise Server

Creating Siebel Enterprise Server Named Subsystems

Creating Siebel Enterprise Server Named Subsystems

This topic describes how to create Siebel Enterprise Server named subsystems.

The named subsystem parameters override the parameters that are set at the Enterprise Server, Siebel Server, and server component levels.

You can create and configure the named subsystems by using either the Siebel Server Manager GUI or the command-line interface.

To create named subsystems

1. Navigate to the Administration - Server Configuration screen, then the Enterprises view.
2. Select the Profile Configuration view tab.
3. In the Profile Configuration list, click New.
    a. In the Profile field, type the name of the named subsystem (profile). Do not exceed 30 characters when defining the name of the named subsystem.

b. In the Alias field, type the alias of the named subsystem.
c. In the Subsystem Type field, click the select button and then select the Subsystem Type from the dialog box and click OK.

   The subsystem type that you select must have a check mark in the Is Named Enabled field.
d. In the Description field, type a description of the named subsystem.
e. Click Menu, and then Save Record.

4. In the Profile Parameters list, modify the parameters as appropriate.

   Parameters are added to the named subsystem, based on the specified subsystem type.

Related Topics

About Siebel System Parameters

About Named Subsystem Parameters

Configuring the Siebel Enterprise Server

Configuring Siebel Enterprise Server Named Subsystem Parameters

Named Subsystem Management Commands

Configuring Siebel Enterprise Parameters

This topic describes how to configure Siebel Enterprise parameters.

To configure Siebel Enterprise parameters

1. Navigate to the Administration - Server Configuration screen, then the Enterprises view.
2. Select the Parameters view tab.
3. In the Enterprise Parameters list, select the parameter of interest and configure its value.

Related Topics

About Siebel System Parameters

Configuring the Siebel Enterprise Server

Configuring Siebel Enterprise Component Definition Parameters

This topic describes how to configure Siebel Enterprise Server component definition parameters.

To configure Siebel Enterprise Server component definition parameters

1. Navigate to the Administration - Server Configuration screen, then the Enterprises view.
2. Select the Component Definitions view tab.
3. In the Component Definitions list, select the Siebel Server component definition of interest.
4. In the Component Parameters list, make any changes to the component parameters.
5. If the component definition is based on a batch mode component, then synchronize the component.

6. If a parameter value is effective at server restart, then restart the Siebel Server for changes to take effect.

The values of fixed parameters can be changed during component reconfiguration. Fixed parameters cannot be changed after the component has been activated or enabled.

Related Topics

About Siebel System Parameters

Configuring the Siebel Enterprise Server

Synchronizing Components on a Siebel Enterprise Server

About Siebel Server Component Definitions

Reconfiguring Siebel Server Component Definitions

About System Alert Notification

System alert notification is a feature that allows a running server component to alert the administrator, using preconfigured communication channels, to any problems that cannot be handled by the server component.

The system alert notification process starts when a server component that has been configured for component notification encounters a problem. This component sends a request to the Siebel Administrator Notification component (alias AdminNotify) with details on the encountered problem and with an appropriate message to send to the administrator. The AdminNotify component then alerts the administrator by using the preconfigured communication channels, for example, email.

AdminNotify is a batch mode, multithreaded server component and is part of the Auxiliary System Management (alias SystemAux) component group. This server component is enabled by default.

Some tasks involve configuring Siebel Server components for notification by using the Server Manager command-line interface.

Note the following points about system alert notification:

- A server component event or task event triggers a system alert notification.
- Losing database connectivity does not trigger a system alert notification.
- If a Siebel Server that hosts an AdminNotify server component is forced to shut down, then the AdminNotify server component also shuts down, resulting in the loss of the system alert notifications for server components or tasks on that Siebel Server.
- A server component or task on a Siebel Server cannot trigger an alert on an AdminNotify server component that is hosted by another Siebel Server in the Siebel Enterprise Server.
- A change in the state of a component (for example, from running to online) does not trigger a system alert notification.
- An administrator shutting down a server component does not trigger an alert.
- The server components required for the communications channel that delivers the system alert notification must be enabled.

This topic contains the following information:

- *Configuring System Alert Notification*

Related Topics

Configuring System Alert Notification

Troubleshooting System Alert Notification

System Alert Notification Commands

Configuring System Alert Notification

This topic describes how to configure system alert notification. It contains the following information:

- *Creating a System Alert Profile*
- *Configuring Server Components to Use System Alerts*

Related Topics

About Siebel System Parameters

About Advanced and Hidden Parameters

Configuring Siebel Enterprise Parameters

About System Alert Notification

Configuring System Alert Notification

Troubleshooting System Alert Notification

Configuring Siebel Server Parameters

Creating a System Alert Profile

Use the following procedure for creating a system alert profile.

To create a system alert profile

1. Navigate to the Administration - Server Configuration screen, then the Enterprises view.
2. Select the System Alerts view tab.
3. In the System Alerts view, click New to create a system alert profile record.
4. Enter values for system alert profile Name, Alias, and Description.
5. Click the Media column drop-down list and select the system alert notification medium.
6. Click Menu, and then Save Record.
7. In the Alert Parameters list, enter values to define the communication parameters. For example, define the email addresses and email server if the notification medium is by email notification.

Related Topics

Configuring System Alert Notification

Configuring Server Components to Use System Alerts

Use the following procedure for configuring server components to use system alerts by setting applicable server component parameters.

To configure components to use system alerts

1. Navigate to the Administration - Server Configuration screen, then the Enterprises view.
2. Select the Component Groups view tab.
3. In the Component Groups list, select the Siebel Server component group of interest.
4. In the Components list, select the server component of interest.
5. In the Component Parameters list, query for and set the parameters described in the following table.

Parameter Name	Parameter Alias	Description
Notification Handler	NotifyHandler	The name of the system alert profile, which specifies the notification medium and settings for the component alerts. Set this value to the alias name of the system alert profile that you defined previously.
Disable Notification	DisableNotification	A Boolean value that enables or disables server component notification. The default value is False.
Time to Wait for doing Notification	NotifyTimeOut	Specifies the amount of time to wait in milliseconds for connecting to the named pipe, which is a system element used in notification communications. If the named pipe does not reply within the specified time, then the notification is terminated. The default value is 100 milliseconds.
Notification Action on Task Exit	NotifyOnTaskExit	An integer value that determines if notifications are sent in case of error. A value of 0 disables error notifications. A value of 1 enables error notifications. The default value is 0. Set this parameter to 1 to configure system alerts. **Note:** This parameter is an advanced parameter, which might not be set for display in the GUI.

Troubleshooting System Alert Notification

This topic provides guidelines for resolving problems with system alert notification. To resolve a problem, look for it in the list of symptoms or error messages in the following table.

Problem	Cause	Solution
Not receiving system alerts	Incorrect enterprise, Siebel Server, or server component configurations	Review information in *Checking Your Siebel Enterprise and Siebel Server Configurations*.
	SMTP server defined in the system alert profile is not	Review your SMTP server documentation to configure these requests.

Problem	Cause	Solution
	configured to accept requests from the Siebel Server	
	Review log files	Review log files of server component AdminNotify for other error messages or conditions. For information about configuring and reading server component log files, see *Siebel System Monitoring and Diagnostics Guide* .

Related Topics

About System Alert Notification

Configuring System Alert Notification

Checking Your Siebel Enterprise and Siebel Server Configurations

Configuring Component Job Templates

This topic describes how to configure a component job template, which you can use on Siebel Servers across an enterprise. A component job template is a predefined component job that uses parameter values that you have defined. Use component job templates instead of individual component jobs if you are planning to regularly run component jobs with the same parameter values.

To define a component job template

1. Navigate to the Administration - Server Configuration screen, then the Job Templates view.
2. In the Job Templates list, click New.
3. In the Name field, type in a descriptive name for the component job, such as Monthly EIM.
4. In the Short Name field, type in an alias for the component job, such as MonthEIM.
5. In the Component field, select the component for this component job, such as Enterprise Integration Manager.

   > **Note:** After a component job is created, do not change the value of the Component field. To change the component for an existing component job, create a new component job instead of modifying the existing one.

6. In the Description field, type in a description of the component job.
7. Click Menu, and then Save Record.
8. In the Job Parameters list, click New.
9. In the Name field, select the parameter that you want to define for this component job.
   a. In the Name field, click the select button.

      The Component Parameters dialog box appears. The parameters that appear in the Component Parameters dialog box vary depending on the component that you specified in an earlier step.
   b. In the Component Parameters dialog box, click Query.
   c. In the Name field, type in the name of the parameter and click Go.
   d. If the query matches the parameter of interest, then click OK.

10. In the Value field of the Job Parameters list, type in the value for the parameter.

    The default value is automatically displayed in this field.
11. Check the appropriate flags for this parameter. To set the parameter type, use the following flags:
    - For a fixed parameter, check the Fixed field.
    - For a required parameter, check the Required field.
12. Click Menu, and then Save Record.
13. Continue to add parameters until you have defined the parameters for the component job.

Related Topics

Configuring the Siebel Enterprise Server

Administering Component Jobs

Synchronizing Components on a Siebel Enterprise Server

This topic describes how to synchronize components on a Siebel Enterprise Server. You must synchronize batch mode Siebel Server components between the Siebel Gateway and the database whenever you:

- Create new component definitions
- Modify existing batch mode component definitions
- Delete Siebel Server components

Note: If synchronization does not take place, then make sure that the LOV (List of Value) type SRM_ACTION_TYPE is set to active. For more information about working with LOVs, see *Siebel Applications Administration Guide* .

To synchronize components on a Siebel Enterprise Server

1. Navigate to the Administration - Server Configuration screen, then the Enterprises view.
2. Select the Synchronize view tab.

   A list of batch mode server components appears.
3. Click Synchronize.

   This operation might take up to a minute to execute.
4. For the changes to take effect, stop and restart the Siebel Server system service.

Related Topics

Configuring the Siebel Enterprise Server

Administering the Siebel Server System Service

Configuring the Siebel Server

This topic lists the configuration tasks and processes applicable to the Siebel Server. For more information about the Siebel Server and the overall system architecture, see *Siebel Enterprise Server Architecture*.

This topic contains the following information:

- *Enabling Component Groups on a Siebel Server*
- *Disabling Component Groups on a Siebel Server*
- *About Starting Siebel Server Components*
- *Automatically Starting a Component on a Siebel Server*
- *Manually Starting or Disabling a Component on a Siebel Server*
- *Configuring Siebel Server Parameters*
- *Configuring Siebel Server Component Parameters*

Enabling Component Groups on a Siebel Server

This topic describes how to enable Siebel Server component groups on an individual Siebel Server.

When you initially configure a Siebel Server, you specify which component groups to enable. When you have to change which groups are enabled or disabled, use the procedures described in this topic and the topic that follows. For more information about the initial configuration tasks for the Siebel Server, see *Siebel Installation Guide* .

To enable a component group on a Siebel Server

1. Navigate to the Administration - Server Configuration screen, then the Enterprises view.
2. Select the Component Groups view tab.
3. In the Component Groups list, select the Siebel Server component group of interest.
4. In the Component Groups Assignments list, select the Siebel Server of interest.
5. Click Enable.
   The Enabled on Server? field of the Siebel Server record becomes checked.
6. For the change to take effect, stop and restart the Siebel Server system service.

Related Topics

About Enabled and Disabled Component Groups

Enabling Component Groups on a Siebel Enterprise Server

Configuring the Siebel Server

Administering the Siebel Server System Service

Disabling Component Groups on a Siebel Server

This topic describes how to disable Siebel Server component groups on an individual Siebel Server.

To disable a component group on a Siebel Server

1. Navigate to the Administration - Server Configuration screen, then the Enterprises view.
2. Select the Component Groups view tab.
3. In the Component Groups list, select the Siebel Server component group of interest.
4. In the Component Groups Assignments list, select the Siebel Server of interest.
5. Click Disable.

   The Enabled on Server? field of the Siebel Server record becomes clear.
6. For the change to take effect, stop and restart the Siebel Server system service.

Related Topics

About Enabled and Disabled Component Groups

Disabling Component Groups on a Siebel Enterprise Server

Configuring the Siebel Server

Administering the Siebel Server System Service

About Starting Siebel Server Components

When a component group is enabled, all of the server components within the component group are started and assigned to the Siebel Servers. The Siebel Servers are configured to use the component group provided that the server components are configured to start automatically. If the server components are not configured to start automatically, then you must start them manually.

You determine the state of a server component by viewing the value that appears in the State field for a component. This field is available, for example, when you choose Administration - Server Management, then Enterprises. The State field for a component can have one of the following possible values:

- **Running.** The server component is online and accepting requests, and at least one task is running.
- **Online.** The server component is online and awaiting a request.
  - For a multithreaded component, if the number of active running processes are greater than the value of the parameter MinMTServers, and no tasks are running for the component, then the state is Online.
  - For a background mode component, if the number of active running processes is less than the value of the parameter DfltTasks, and no tasks are running for the component, then the state is Online.
- **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.
- **Unavailable.** The server component is unavailable.

- **Paused.** The server component is online but is not accepting new requests.
- **Shutting down.** The server component is shutting down and cannot accept new requests. At least one task is still running.
- **Shutdown.** The server component is shut down.

Related Topics

Automatically Starting a Component on a Siebel Server

Manually Starting or Disabling a Component on a Siebel Server

Automatically Starting a Component on a Siebel Server

This topic describes how to configure a Siebel Server component to start automatically when the Siebel Server starts.

To automatically start a component on a Siebel Server

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. In the Components list, select the server component that you want to configure.
4. Click Auto Start.

   The server component starts automatically when the Siebel Server starts.

Related Topics

Configuring the Siebel Server

About Starting Siebel Server Components

Manually Starting or Disabling a Component on a Siebel Server

Manually Starting or Disabling a Component on a Siebel Server

This topic describes how to configure a Siebel Server component so that you must manually start it after the Siebel Server starts.

To manually start a component on a Siebel Server

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. In the Components list, select the server component that you want to configure.
4. Click Manual Start.

   The server component requires that you start it after the Siebel Server starts.

Related Topics

Configuring the Siebel Server

About Starting Siebel Server Components

Automatically Starting a Component on a Siebel Server

Configuring Siebel Server Parameters

This topic describes how to configure Siebel Server parameters.

To configure Siebel Server parameters

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. Select the Parameters view tab.
4. In the Parameters list, select the parameter of interest and configure its value.

Related Topics

About Siebel System Parameters

About Siebel Server Parameters

Configuring the Siebel Server

Configuring Siebel Server Component Parameters

This topic describes how to configure Siebel Server component parameters.

To configure Siebel Server component parameters

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, then select the Parameters view tab.
5. In the Component Parameters list, select the parameter of interest and configure its value.

Related Topics

About Siebel System Parameters

About Siebel Component Parameters

Configuring the Siebel Server

Checking Your Siebel Enterprise and Siebel Server Configurations

One of the main objectives of your Siebel Enterprise and Siebel Server configurations is to make sure that server components are properly configured and ready to execute tasks. Check the following items to make sure that you meet the necessary configuration conditions for this objective:

- The component group that contains the server component member is enabled at the enterprise level.
- The component group that contains the server component is enabled at the Siebel Server level.
- The component definition is enabled. The component definition defines the configured component for the Siebel Enterprise and for all of its Siebel Servers.

If the previously listed conditions are true, then the Siebel Server allocates resources (some space in the shared memory and a public port) for the server component to use when the Siebel Server starts. The Siebel Server also marks the server component as ready (available) to start new tasks and performs any additional steps necessary, for example, starting shells if the server component is multithreaded. The number of multithreaded shells started is governed by the parameter Minimum MT Servers (alias MinMTServers). If the server component is a background mode component, then it starts the background number of tasks governed by the parameter Default Tasks (alias DfltTasks).

Related Topics

Enabling Component Groups on a Siebel Enterprise Server

Enabling Component Groups on a Siebel Server

Activating a Custom Siebel Server Component Definition

About Siebel Server Component Definitions

If you want to use customized components, then you can create defined components or customize existing components. After it is defined, a component can have one of three definition states: Creating, Active, or Inactive.

- **Creating.** Indicates that the defined component is being configured. After the definition is configured, activating the component definition fixes its configuration, changes the component's state to Active, and allows the component to be assigned to Siebel Servers. The fixed parameters for the defined component cannot be overridden when the component is assigned, or when tasks are started for the component.
- **Active.** Indicates that the defined component definition state is available for registration on Siebel Servers.
- **Inactive.** Indicates that the defined component will be inactivated when you restart the Siebel Server (or servers) to which the component is assigned. The component remains assigned to the Siebel Servers, but tasks cannot be started for the component until you revert the component definition state to Active and restart the Siebel Servers.

Parameter values in a component definition are used to initialize the component on a specific Siebel Server.

> **Note:** If component definitions are created, modified, or deleted for batch mode components, then it is necessary to synchronize the components with the Siebel Gateway.

Related Topics

Configuring Siebel Enterprise Component Definition Parameters

Synchronizing Components on a Siebel Enterprise Server

Process of Creating a Custom Siebel Server Component

Deleting a Siebel Server Component Definition

Deactivating a Siebel Server Component Definition

Process of Creating a Custom Siebel Server Component

This topic describes the process of creating a custom Siebel Server component.

To create a custom Siebel Server component

1. (Optional) Create a new component group.

   Do not perform this task if you plan to add the new custom Siebel Server component to an existing component group. For information about this task, see *Creating a Custom Siebel Server Component Group*.
2. Create a new component definition.

   For information about this task, see *Creating a Custom Siebel Server Component Definition*.
3. Activate the new component definition.

   For information about this task, see *Activating a Custom Siebel Server Component Definition*.
4. Synchronize components if the new custom Siebel Server component is a batch mode component.

   For information about this task, see *Synchronizing Components on a Siebel Enterprise Server*.
5. Stop and restart the Siebel Server system service.

   For information about this task, see *Administering the Siebel Server System Service*.

Related Topics

About Siebel Server Component Definitions

Deleting a Siebel Server Component Definition

Deactivating a Siebel Server Component Definition

Creating a Custom Siebel Server Component Group

If you want to create your own defined components and assign them to component groups other than the predefined ones, then you first must create component groups before creating the defined components. Component groups allow you to run related tasks and administer related components in logical groupings. Do not perform this task if you plan to add a new custom Siebel Server component to an existing component group.

This task is an optional step in *Process of Creating a Custom Siebel Server Component*.

To create a custom component group

1. Navigate to the Administration - Server Configuration screen, then the Enterprises view.
2. Select the Component Groups view tab.
3. In the Component Groups list, click New.
4. In the Name field, type in a name for the component group.

   The name must be unique across the Siebel Enterprise Server. Specify a name that expressively identifies the component group.
5. In the Alias field, type in an alias for the component group.

   The component group alias must:

   - Be unique across the Siebel Enterprise Server
   - Not contain any spaces

   The maximum alias length is dependent on the operating system.
6. In the Description field, enter a description of the component group.
7. Click Menu, and then Save Record.

Creating a Custom Siebel Server Component Definition

This topic describes how to create a custom Siebel Server component definition. Each custom server component definition is based on a Siebel Server component type.

This task is a step in *Process of Creating a Custom Siebel Server Component*.

To create a custom Siebel Server component definition

1. Navigate to the Administration - Server Configuration screen, then the Enterprises view.
2. Select the Component Definitions view tab.
3. In the Component Definitions list, click Menu, and then New Record.
4. In the Component field, type in a name for the component.

   The component name must:

   - Be unique across Siebel Enterprise Servers
   - Expressively identify the defined component
   - Not contain any numbers
5. In the Alias field, type in an alias for the component.

   The component alias must:

   - Be unique across Siebel Enterprise Servers
   - Not contain any spaces

   The maximum alias length is dependent on the operating system.
6. In the Component Type field, click the select button and choose the component type that you want to use as the template for this component.

7. In the Description field, type in a description of this component.
8. In the Component Group field, click the select button and choose the component group to which this component will belong.
   The group must exist before you can select it.
   Note: You cannot modify the specified component group after you modify the component record.
9. Click Menu, and then Save Record.
   The State field changes to Creating.
10. In the Component Parameters list, make any changes to the component parameters that were created.

The values of fixed parameters can be changed during component reconfiguration. Fixed parameters cannot be changed after you activate the component.

Related Topics

About Siebel Server Component Definitions

Reconfiguring Siebel Server Component Definitions

Activating a Custom Siebel Server Component Definition

This topic describes how to activate a custom Siebel Server component definition.

This task is a step in *Process of Creating a Custom Siebel Server Component*.

To activate a custom Siebel Server component definition

1. Navigate to the Administration - Server Configuration screen, then the Enterprises view.
2. Select the Component Definitions view tab.
3. In the Component Definitions list, select the component definition of interest (that is not currently activated).
4. Click Activate.
   The component definition state field changes to Active.
5. If the component definition is based on a batch mode component, then synchronize the component.
6. For the change to take effect, stop and restart the Siebel Server system service.

Related Topics

Synchronizing Components on a Siebel Enterprise Server

About Siebel Server Component Definitions

Administering the Siebel Server System Service

Deleting a Siebel Server Component Definition

This topic describes how to delete a Siebel Server component definition.

CAUTION: Make sure that no active server components use the component definition that you want to delete. If you are in doubt, then deactivate the component definition rather than delete it.

To delete a Siebel Server component definition

1. Navigate to the Administration - Server Configuration screen, then the Enterprises view.
2. Select the Component Definitions view tab.
3. In the Component Definitions list, select the component definition of interest.
4. Click Delete.

Related Topics

About Siebel Server Component Definitions

Process of Creating a Custom Siebel Server Component

Deactivating a Siebel Server Component Definition

Deactivating a Siebel Server Component Definition

This topic describes how to deactivate a custom Siebel Server component definition.

To deactivate a Siebel Server component definition

1. Navigate to the Administration - Server Configuration screen, then the Enterprises view.
2. Select the Component Definitions view tab.
3. In the Component Definitions list, select the component definition of interest (that is currently activated).
4. Click Deactivate.
   The State field for the component definition changes to Inactive.
5. For the change to take effect, stop and restart the Siebel Server system service.

Related Topics

About Siebel Server Component Definitions

Process of Creating a Custom Siebel Server Component

Deleting a Siebel Server Component Definition

Administering the Siebel Server System Service

Advanced Configuration Tasks

This topic lists advanced configuration tasks and processes applicable to the Siebel Enterprise Server and Siebel Server. For more information about the Siebel Server and the overall system architecture, see *Siebel Enterprise Server Architecture* .

For common configuration tasks for the Siebel Enterprise Server and Siebel Server, see:

- *Configuring the Siebel Enterprise Server*
- *Configuring the Siebel Server*

This topic contains the following information:

- *Reconfiguring Siebel Server Component Definitions*
- *Configuring Automatic Restart for Server Components*
- *Configuring Database Reconnect for Server Components*
- *Configuring Memory-Based Server Component Recycling*
- *Deleting System Parameter Overrides*

Reconfiguring Siebel Server Component Definitions

Component reconfiguration is a process that allows existing component tasks to continue running until they finish, while starting new processes with reconfigured component parameter values. Each component can be reconfigured, but this task is primarily done for multithreaded components. This topic is part of *Advanced Configuration Tasks*.

Component reconfiguration is particularly useful for scenarios such as site migration. In this scenario, component reconfiguration allows you to maintain the existing component configuration, which remains available until the migration is complete and a new component configuration becomes available.

When a component reconfiguration is committed, this action signifies to every Siebel Server in the Siebel Enterprise Server that a new component definition is available and that new tasks should not start for that component. Existing tasks, however, continue to run until completion. The connected Siebel Servers then start new multithreaded processes for the component, using the new parameter values that were read from the Siebel Gateway registry. On each Siebel Server, the number of new multithreaded processes is governed by the parameter Minimum MT Servers (alias MinMTServers).

Previous component tasks that are not affected by the component reconfiguration can continue to run for some time. Therefore, the multithreaded processes hosting the tasks continue to run and the tasks occupy slots in shared memory. Because old and new multithreaded processes for the component are both running, there might be an increase in the number of multithreaded processes running on the Siebel Server computers.

Therefore, it is recommended that you run component definition reconfiguration at times when the server component workload is low, that is, when the component's existing tasks or sessions are few in number. Doing so allows the existing tasks or sessions on the old multithreaded processes to end, so that the total numbers of multithreaded processes and tasks do not exceed configured limits or exhaust system resources. When all of its tasks have ended, a multithreaded process eventually ends.

For each Siebel Server, initial sizing would have been performed, as described in *Siebel Performance Tuning Guide* . The total number of multithreaded processes for the component cannot exceed what is specified using the parameter Maximum MT Servers (alias MaxMTServers) and the total number of tasks cannot exceed what is specified using the parameter Maximum Tasks (alias MaxTasks).

> **Note:** Parameters set at the individual component level are not affected by component definition reconfiguration unless the parameter override is deleted on that parameter.

Before reconfiguring Siebel Server components, make sure that:

- All of the Siebel Servers running the server component designated for reconfiguration are running. This check makes sure that the server component instances receive the reconfigured parameter values.

- Any external resources that are accessed by current tasks running on the existing component configuration remain available.
  Examples of external resources include the configuration files that are defined in the component configuration.
- The Siebel runtime repository in effect reflects the application configuration you require.
- Any external resources that are defined in the new component configuration are available.
- Any previous component reconfiguration process has already completed. If it has not, then you might have to wait before you can reconfigure again.
  If existing multithreaded processes are still running along with reconfigured multithreaded processes, then the following error message might appear: `Unable to complete reconfiguration due to insufficient process resources.` Starting another reconfiguration process at this time would require more reconfigured multithreaded processes on all applicable Siebel Servers, which might exceed the limits that you configured during sizing and tuning.

To reconfigure Siebel Server components

1. Navigate to the Administration - Server Configuration screen, then the Enterprises view.
2. Select the Component Definitions view tab.
3. In the Component Definitions list, select the component definition that you want to reconfigure.
4. Click Menu, and then Start Reconfiguration.
   The Definition State field changes to Reconfiguring.
5. In the lower Component Definitions list, change the Value field of parameters that you want to reconfigure for the component.
   You can also change the values of fixed parameters, but you cannot change whether parameters are fixed.
6. After parameter values have been reconfigured, commit the new configuration by clicking Menu, and then Commit Reconfiguration.
   The new parameter values are merged at the enterprise level. To cancel the reconfiguration before it has been committed, click Menu, and then Cancel Reconfiguration.

Related Topics

Reconfiguring Component Definition Commands

Related Books

Siebel Performance Tuning Guide

Configuring Automatic Restart for Server Components

Automatic restart is a feature that allows a Siebel Server component to automatically attempt a restart if the component exits with an error. For a user-defined number of times after the error, the Siebel Server tries to restart the component. This feature greatly reduces the administration of Siebel Server components. By default, this feature is disabled for all components. This topic is part of *Advanced Configuration Tasks*.

> **Note:** If a Siebel Server component is terminated, that is, it does not exit with an error, then the Siebel Server component is not restarted.

You can configure automatic restart by using the following parameters:

- Auto Restart (alias AutoRestart)
- Minimum Up Time (alias MinUpTime)
- Number of Restarts (alias NumRestart)

The Minimum Up Time and Number of Restarts parameters combine to determine the number of restart attempts in a time interval allowed for a component (NumRestart multiplied by MinUpTime). If a component instance cannot be restarted after this time interval, then no new restart is attempted (therefore, the component instance will not run).

Typically, you set these parameters at the component level, but, depending on your system configuration, you might want to set these parameters at the Siebel Enterprise Server or Siebel Server levels.

Related Topics

About Siebel System Parameters

Siebel Enterprise, Server, and Component Parameters

Configuring Database Reconnect for Server Components

Database reconnect is an optional feature that enables Siebel Server components to automatically attempt a database connection following a database or network failure. At regular intervals after the failure, Siebel Server components try to reconnect to the database and resume any tasks that were rolled back due to the failure. This feature greatly reduces the administration of Siebel Server components. This topic is part of *Advanced Configuration Tasks*.

Without database reconnect, a database or network failure causes the Siebel Server component to shut down and all of the running tasks to crash. You must then manually restart all of the components that were shut down and rerun all of the tasks that crashed.

Database reconnect is enabled for all of the background mode and batch mode Siebel Server components, except for the Enterprise Integration Manager and Database Extract components. This feature is disabled for all of the interactive mode Siebel Server components (such as Synchronization Manager and all of the Application Object Manager components).

You can configure database reconnect by using the following parameters:

- Number of Retries (alias NumRetries)
- Retry Interval (alias RetryInterval)
- Retry Up Time (alias RetryUpTime)

Set these parameters at the same level as the automatic restart parameters (typically, at the component level).

Related Topics

About Siebel System Parameters

Configuring Automatic Restart for Server Components

Siebel Enterprise, Server, and Component Parameters

Configuring Memory-Based Server Component Recycling

If certain multithreaded server components within your Siebel application are experiencing excessive memory consumption, then you can configure certain parameters to configure a component process to restart automatically. This feature, called component recycling, allows continued operation of server components without affecting end users. This topic is part of *Advanced Configuration Tasks*.

The memory-based component recycling feature operates as follows:

- Identifies a process for recycling by monitoring virtual memory usage
- Flags the process not to accept any new requests
- Starts another process to take the place of the original process
- Waits for all of the current tasks to complete
- Shuts down the process, which releases memory resources back to the operating system

To configure memory-based recycling of server components, set the parameters Memory Usage Based Multithread Shell Recycling (alias MemoryBasedRecycle) and Process VM Usage Lower Limit (alias MemoryLimit) at the component level for the affected multithreaded server component. The parameter MemoryBasedRecycle allows the recycling feature to monitor the virtual memory of a process and begins the recycling procedure when a certain memory limit is reached. For example, on Windows, this virtual memory is the working set of a process. The parameter MemoryLimit sets the memory limit at which recycling occurs.

You can also set a value in percent for the parameter Process VM Usage Upper Limit (alias MemoryLimitPercent). The default value is 20. When memory usage exceeds the value of MemoryLimitPercent above MemoryLimit, a fast shutdown of the memory consuming process is triggered rather than a normal shutdown.

Set the parameters for memory-based server component recycling at the component level.

For a listing of preconfigured server components, see the table in *Siebel Server Components*, which indicates whether the server component is multithreaded and, therefore, eligible to use the memory-based recycling feature.

Related Topics

Siebel Enterprise, Server, and Component Parameters

Generic Parameters

Deleting System Parameter Overrides

As previously described, lower-level system parameters inherit values from the same higher-level system parameters. However, modifying a lower-level parameter value sets an override at this level. That is, the lower-level parameter loses the ability to inherit values from higher-level parameters. To reinstate the default functionality, perform a delete parameter override operation on affected parameters. This topic is part of *Advanced Configuration Tasks*.

The following procedures describe how to delete parameter overrides for Siebel Enterprise Server parameters, Siebel Server parameters, component definition parameters, and Siebel Server component parameters.

This topic contains the following information:

Deleting a Parameter Override for a Siebel Enterprise Parameter

Deleting a Parameter Override for a Siebel Server Parameter

Deleting a Parameter Override for a Component Definition Parameter

Deleting a Parameter Override for a Siebel Server Component Parameter

Related Topics

About Siebel Server Parameters

Deleting a Parameter Override for a Siebel Enterprise Parameter

This procedure describes how to delete a parameter override for a Siebel Enterprise parameter.

To delete a parameter override for a Siebel Enterprise parameter

1. Navigate to the Administration - Server Configuration screen, then the Enterprises view.
2. Select the Parameters view tab.
3. In the Enterprise Parameters list, select the parameter of interest.
4. Click Menu, and then Delete Parameter Override.

Deleting a Parameter Override for a Siebel Server Parameter

This procedure describes how to delete a parameter override for a Siebel Server parameter.

To delete a parameter override for a Siebel Server parameter

1. Navigate to the Administration - Server Configuration screen, then the Servers view.
2. Select the Parameters view tab.
3. In the Server Parameters list, select the parameter of interest.
4. Click Menu, and then Delete Parameter Override.

Deleting a Parameter Override for a Component Definition Parameter

This procedure describes how to delete a parameter override for a component definition parameter.

To delete a parameter override for a component definition parameter

1. Navigate to the Administration - Server Configuration screen, then the Enterprises view.
2. Select the Component Definitions view tab.
3. In the Component Definitions list, select the component definition that you want to reconfigure.
4. Select the Parameters view tab.
5. In the Component Parameters list, select the parameter of interest.
6. Click Menu, and then Delete Parameter Override.

Deleting a Parameter Override for a Siebel Server Component Parameter

This procedure describes how to delete a parameter override for a Siebel Server component parameter.

To delete a parameter override for a Siebel Server component parameter

1. Navigate to the Administration - Server Configuration screen, then the Servers view.
2. Select the Components view tab.
3. Select the component of interest, then click the Parameters view tab .

4. In the Component Parameters list, select the parameter of interest.
5. Click Menu, and then Delete Parameter Override.

5 Administering Server System Services

Administering Server System Services

This chapter describes the sequence in which to start up and shut down your Siebel deployment, and provides the procedures required to start, stop, and administer the Siebel Gateway system service and the Siebel Server system service.

This chapter includes the following topics:

- *Starting and Shutting Down a Siebel CRM Deployment*
- *Administering the Siebel Gateway System Service*
- *Administering the Siebel Server System Service*
- *Renaming Servers for an Existing Siebel CRM Deployment*

Starting and Shutting Down a Siebel CRM Deployment

It is important to follow the correct sequence when starting or shutting down a Siebel CRM deployment, because several dependencies require that certain servers are running before others. Perform the following procedures to start or shut down your Siebel CRM deployment.

This topic contains the following information:

- *Starting a Siebel CRM Deployment*
- *Shutting Down a Siebel CRM Deployment*

Related Topics

Configuring Siebel Server Load Balancing

Administering the Siebel Gateway System Service

Administering the Siebel Server System Service

Shutting Down a Siebel Server

Siebel Server Manager Commands

Backing Up and Restoring the Siebel Gateway Registry

Related Books

Siebel Deployment Planning Guide

Siebel Installation Guide

Siebel Security Guide

Starting a Siebel CRM Deployment

This procedure describes how to start a Siebel CRM deployment. This topic is part of *Starting and Shutting Down a Siebel CRM Deployment*.

To start a Siebel CRM deployment

1. If necessary, start the Siebel database.
   For more information about this task, see your database documentation.
2. Start the Siebel Gateway system service.
   For more information, see *Administering the Siebel Gateway System Service*.
3. Start the application container for each applicable installed module.
   For more information about stopping and starting the application container, and about configuring autostart for the application container on UNIX operating systems, see *Siebel Installation Guide* .
4. Start any third-party software, if applicable.
5. Start the Siebel Server system service.
   The Siebel Server must connect to the Siebel Gateway and Siebel database on startup.
   For more information, see *Administering the Siebel Server System Service*.

Shutting Down a Siebel CRM Deployment

This procedure describes how to shut down a Siebel CRM deployment. This topic is part of *Starting and Shutting Down a Siebel CRM Deployment*.

To shut down a Siebel CRM deployment

1. Shut down the Siebel Server system service.
   When the Siebel Server system service is shut down, it shuts down server components and tasks before shutting down itself.
   For more information, see *Administering the Siebel Server System Service*.

   > **Note:** To make sure that server components shut down properly, shut down Siebel Servers before shutting down the Siebel Server system service.

2. Shut down the application container for each applicable installed module.
   For more information about stopping and starting the application container, and about configuring autostart for the application container on UNIX operating systems, see *Siebel Installation Guide* .
3. Shut down the Siebel Gateway system service.
   For more information, see *Administering the Siebel Gateway System Service*.

   > **Note:** Make sure that all of the Siebel Servers are shut down before shutting down the Siebel Gateway service.

4. Shut down the Siebel database.

   For more information about this task, see your database documentation. This step is not necessary if you are restarting your Siebel CRM deployment.

Administering the Siebel Gateway System Service

Occasionally, you must stop and restart the Siebel Gateway system service for maintenance purposes. Restart the system service only when it is necessary. Running the Siebel Gateway system service starts the Siebel Gateway registry.

> **Note:** For compatibility with prior releases, the actual name of the system service is Siebel Gateway Name Server. However, this guide sometimes refers instead to Siebel Gateway, for consistency with current usage.

The Siebel Gateway authenticates any client attempting to access configuration information. You set up the method of authentication during initial configuration. For more information, see *About Siebel Gateway Authentication*.

Administering the Siebel Gateway also sometimes requires you to start or stop the application container manually, as noted in *Starting and Shutting Down a Siebel CRM Deployment*. See also *Siebel Installation Guide* .

This topic contains the following information:

- *About Siebel Gateway Authentication*
- *Administering the Siebel Gateway System Service on Windows*
- *Administering the Siebel Server System Service on UNIX*

Related Topics

About the Siebel Gateway

Starting and Shutting Down a Siebel CRM Deployment

Related Books

Siebel Installation Guide

Siebel Security Guide

About Siebel Gateway Authentication

This topic provides an overview of Siebel Gateway authentication. All of the server and client side executable programs that connect to the Siebel Gateway, including the Siebel Server, pass a user name and password to the Siebel Gateway for authentication.

Configuration of Siebel Gateway authentication is part of initial configuration using Siebel Management Console. No additional configuration is required, unless, for instance, you change a data source, or choose to use a different means of authentication than that specified during initial configuration. For more information, see *Siebel Installation Guide* and *Siebel Security Guide* . This topic is part of *Administering the Siebel Gateway System Service*.

Administering the Siebel Gateway System Service on Windows

This topic describes how to start, stop, and check the status of the Siebel Gateway system service on Windows. This topic is part of *Administering the Siebel Gateway System Service*.

This topic contains the following information:

- *Starting the Siebel Gateway System Service on Windows*
- *Stopping the Siebel Gateway System Service on Windows*
- *Checking the Status of the Siebel Gateway System Service on Windows*

Starting the Siebel Gateway System Service on Windows

This topic describes how to start the Siebel Gateway system service on Windows.

To start the Siebel Gateway system service on Windows

1. Right-click My Computer.
2. Click Manage.
   The Computer Management panel appears.
3. In the Computer Management tree, expand Services and Applications.
4. Click Services.
5. In the details panel, scroll through the list of services and select Siebel Gateway Name Server.
6. Right-click Siebel Gateway Name Server and select Start.
   Windows starts the Siebel Gateway Name Server system service. This operation might take a few seconds. After the service has started, the Status field changes to Started.
7. If the application container is not running, then start it manually.
   For more information, see *Starting and Shutting Down a Siebel CRM Deployment*.

Stopping the Siebel Gateway System Service on Windows

This topic describes how to stop the Siebel Gateway system service on Windows.

To stop the Siebel Gateway system service on Windows

1. If the application container is running, then stop it manually.
   For more information, see *Starting and Shutting Down a Siebel CRM Deployment*.
2. Right-click My Computer.
3. Click Manage.
   The Computer Management panel appears.
4. In the Computer Management tree, expand Services and Applications.
5. Click Services.
6. In the details panel, scroll through the list of services and select Siebel Gateway Name Server.
7. Right-click Siebel Gateway Name Server and select Stop.
   The Stop Other Services dialog box now appears and asks whether you want to stop the Siebel Server system service.

8. Click Yes.

   Windows stops the Siebel Gateway Name Server system service and the Siebel Server system service. This operation might take a few seconds. After the services have stopped, the Status field is blank.

Checking the Status of the Siebel Gateway System Service on Windows

This topic describes how to check the status of the Siebel Gateway system service on Windows.

To check the status of the Siebel Gateway system service on Windows

1. Right-click My Computer.
2. Click Manage.

   The Computer Management panel appears.
3. In the Computer Management tree, expand Services and Applications.
4. Click Services.
5. In the details panel, scroll through the list of services and select Siebel Gateway Name Server.

   A value of Started in the Status field for the specified service indicates that the Siebel Gateway Name Server system service is running. If the Status field is blank, then the system service is not currently running.
6. Check whether the application container is running.

   For more information, see *Starting and Shutting Down a Siebel CRM Deployment*.

Administering the Siebel Gateway System Service on UNIX

This topic describes how to start, stop, and check the status of the Siebel Gateway system service on UNIX. This topic is part of *Administering the Siebel Gateway System Service*.

This topic contains the following information:

- *Starting the Siebel Gateway System Service on UNIX*
- *Stopping the Siebel Gateway System Service on UNIX*
- *Checking the Status of the Siebel Gateway System Service on UNIX*

Starting the Siebel Gateway System Service on UNIX

This procedure describes how to start the Siebel Gateway system service on UNIX.

To start the Siebel Gateway system service on UNIX

1. On the Siebel Gateway computer, log in as the Siebel Service owner user.
2. Run the siebenv.sh or siebenv.csh script to set Siebel environment variables.

   For more information about these scripts, see *Siebel Installation Guide* .
3. To start the Siebel Gateway, enter a command like the following:

   ```
   start_ns
   ```

   The following are explanations of some of the available command-line flags:

- To specify the Siebel root directory, use the -r flag, as follows:

  ```
  start_ns -r siebel_root
  ```

  Typically, you do not have to use the -r flag, because the script in the previous step sets the SIEBEL_ROOT environment variable. If the variable is not set, then you must specify the Siebel root directory, using this flag, to indicate the Siebel installation under which the Siebel Gateway runs.
- To start the Siebel Gateway only if currently marked with the autostart attribute, use the -a flag, as follows:

  ```
  start_ns -a
  ```

  Typically, use the -a flag only when invoking the `start_ns` script from an autostart script. For more information about the autostart script, see *Siebel Installation Guide* .
- To force the startup, use the -f flag, as follows:

  ```
  start_ns -f
  ```

  This command can be used to make sure that the Siebel Gateway starts even if it was not previously shut down completely. This flag is typically not needed.

4. Run the `ps` command and check whether the application container for the Siebel Gateway (javaw process daemon) is running. Start it if necessary.

   For more information, see *Starting and Shutting Down a Siebel CRM Deployment*.

Stopping the Siebel Gateway System Service on UNIX

This procedure describes how to stop the Siebel Gateway system service on UNIX.

To stop the Siebel Gateway system service on UNIX

1. On the Siebel Gateway computer, log in as the Siebel Service owner user.
2. Run the siebenv.sh or siebenv.csh script in the current shell process.

   For more information about these scripts, see *Siebel Installation Guide* .
3. Run the `ps` command and check whether the application container for the Siebel Gateway (javaw process daemon) is running. Stop it as necessary.

   For more information, see *Starting and Shutting Down a Siebel CRM Deployment*.

4. To stop the Siebel Gateway system service, enter a command like the following:

```
stop_ns -r siebel_root -f
```

   The following are explanations of some of the available command-line flags:

   - To specify the Siebel root directory, use the -r flag, as follows:

     ```
     stop_ns -r siebel_root -f
     ```

     Typically, you do not have to use the -r flag, because the script in the previous step sets the SIEBEL_ROOT environment variable. If the variable is not set, then you must specify the Siebel root directory, using this flag, to indicate the Siebel installation under which the Siebel Gateway is running.
   - To force the shutdown, use the -f flag, as follows:

     ```
     stop_ns -f
     ```

     This flag causes the Siebel Gateway to shut down sooner, but it might not shut down completely. In general, use the -f flag only if the Siebel Gateway did not respond to the unforced shutdown in a timely manner.

Checking the Status of the Siebel Gateway System Service on UNIX

This procedure describes how to check the status of the Siebel Gateway system service on UNIX.

To check the status of the Siebel Gateway system service on UNIX

1. On the Siebel Gateway computer, log in as the Siebel Service owner user.
2. Run the siebenv.sh or siebenv.csh script to set Siebel environment variables.

   For more information about these scripts, see *Siebel Installation Guide* .
3. To check the status of the Siebel Gateway, enter a command like the following:

```
list_ns
```

   To specify the Siebel root directory, use the -r flag, as follows:

```
list_ns -r siebel_root
```

   Typically, you do not have to use the -r flag, because the script in the previous step sets the SIEBEL_ROOT environment variable. If the variable is not set, then you must specify the Siebel root directory, using this flag, to indicate the Siebel installation under which the Siebel Gateway is configured.
4. Run the `ps` command and check whether the application container for the Siebel Gateway (javaw process daemon) is running.

   For more information, see *Starting and Shutting Down a Siebel CRM Deployment*.

Administering the Siebel Server System Service

Occasionally, you must stop and restart the Siebel Server system service for certain administrative changes to take effect. Restart the system service only when it is necessary.

Note: The Siebel Server system service must be running before any Siebel Server can be started.

Administering the Siebel Server also sometimes requires you to start or stop the application container manually, as noted in *Starting and Shutting Down a Siebel CRM Deployment*. See also *Siebel Installation Guide* .

This topic contains the following information:

- *Administering the Siebel Server System Service on Windows*
- *Administering the Siebel Server System Service on UNIX*

Related Topics

About the Siebel Server System Service

Starting and Shutting Down a Siebel CRM Deployment

Related Books

Siebel Installation Guide

Siebel Security Guide

Administering the Siebel Server System Service on Windows

This topic describes how to start, stop, and check the status of the Siebel Server system service on Windows. This topic is part of *Administering the Siebel Server System Service*.

This topic contains the following information:

- *Starting the Siebel Server System Service on Windows*
- *Stopping the Siebel Server System Service on Windows*
- *Checking the Status of the Siebel Server System Service on Windows*

Starting the Siebel Server System Service on Windows

This procedure describes how to start the Siebel Server system service on Windows.

To start the Siebel Server system service on Windows

1. If the application container is not running, then start it manually.

   For more information, see *Starting and Shutting Down a Siebel CRM Deployment*.
2. Right-click My Computer.
3. Click Manage.

The Computer Management panel appears.

4. In the Computer Management tree, expand Services and Applications.
5. Click Services.
6. In the details panel, scroll through the list of services and select the Siebel Server system service that you need (the enterprise name and Siebel Server name are indicated within square brackets).
7. Right-click Siebel Server and select Start.

   Windows starts the Siebel Server system service. This operation might take a few seconds. After the service has started, the Status field changes to Started.

Stopping the Siebel Server System Service on Windows

This procedure describes how to stop the Siebel Server system service on Windows.

To stop the Siebel Server system service on Windows

1. Right-click My Computer.
2. Click Manage.

   The Computer Management panel appears.
3. In the Computer Management tree, expand Services and Applications.
4. Click Services.
5. In the details panel, scroll through the list of services and select the Siebel Server system service that you need (the enterprise name and Siebel Server name are indicated within square brackets).
6. Right-click Siebel Server and select Stop.

   Windows stops the Siebel Server system service. This operation might take a few seconds. After the service has stopped, the Status field is blank.

   > **Note:** A Microsoft Windows 1053 error might occur during this process. This error does not prevent the Siebel Server system service from stopping.

7. If the application container is running, then stop it manually.

   For more information, see *Starting and Shutting Down a Siebel CRM Deployment*.

Checking the Status of the Siebel Server System Service on Windows

This procedure describes how to check the status of the Siebel Server system service on Windows.

To check the status of the Siebel Server system service on Windows

1. Check whether the application container is running.

   For more information, see *Starting and Shutting Down a Siebel CRM Deployment*.
2. Right-click My Computer.
3. Click Manage.

   The Computer Management panel appears.
4. In the Computer Management tree, expand Services and Applications.
5. Click Services.
6. In the details panel, scroll through the list of services and select the Siebel Server system service that you need (the enterprise name and Siebel Server name are indicated within square brackets).

A value of Started in the Status field for the specified service indicates that the system service is running for the Siebel Server. If the Status field is blank, then the system service is not currently running.

Administering the Siebel Server System Service on UNIX

This topic describes how to start, stop, check, and reset the Siebel Server system service daemon process on UNIX. This topic is part of *Administering the Siebel Server System Service*.

This topic contains the following information:

- *Starting the Siebel Server System Service on UNIX*
- *Stopping the Siebel Server System Service on UNIX*
- *Resetting the Siebel Server System Service on UNIX*

Starting the Siebel Server System Service on UNIX

This procedure describes how to start the Siebel Server system service on UNIX.

To start the Siebel Server system service on UNIX

1. On the Siebel Server computer, log in as the Siebel Service owner user.
2. Run the siebenv.sh or siebenv.csh script to set Siebel environment variables.

   For more information about these scripts, see *Siebel Installation Guide* .
3. Run the `ps` command and check whether the application container for the Siebel Server (javaw process daemon) is running. Start it if necessary.

   For more information, see *Starting and Shutting Down a Siebel CRM Deployment*.
4. Enter the following command:

   ```
   start_server siebel_server_name
   ```

   In this command, siebel_server_name is the name of the Siebel Server.

   You can run this script to start the system service for a specified Siebel Server (or servers), either within a specified Siebel Enterprise Server or across Siebel Enterprise Servers defined for the current installation. With some of the available command-line flags, you can do the following:

   - To start multiple servers, enter the names of the Siebel Servers (separated by spaces), or enter `all` to start all of the Siebel Servers configured under the specified SIEBEL_ROOT on the particular server computer (or all of the Siebel Servers for a particular Siebel Enterprise Server, if the Siebel Enterprise Server name is specified, as described for the use of the -e flag). Use one of the following commands:

     ```
     start_server server1 server2...
     start_server all
     ```

   - To specify the Siebel root directory, use the -r flag, as follows:

     ```
     start_server -r siebel_root
     ```

Typically, you do not have to use the -r flag, because the script in the previous step sets the SIEBEL_ROOT environment variable. If the variable is not set, then you must specify the Siebel root directory, using this flag, to indicate the Siebel installation under which the Siebel Server (or servers) run.

- To limit the operation to Siebel Servers in a specific Siebel Enterprise Server, use the -e flag, as follows:

```
start_server -e enterprise server1 server2...
```

You do not have to use the -e flag if the SIEBEL_ENTERPRISE environment variable is set in the siebenv.sh or siebenv.csh script during initial configuration. If the variable is not set, then you must specify the Siebel Enterprise Server name. To start all of the servers for all of the Siebel Enterprise Servers configured for the SIEBEL_ROOT, do not use the -e flag (you might also have to unset the SIEBEL_ENTERPRISE environment variable).

For example, to start the Siebel Server system services for the prod01 server in the Siebel Enterprise Server, use the -e flag, as follows:

```
start_server -e siebel prod01
```

- To start the Siebel Server system services for the prod01 and prod02 servers in the Siebel Enterprise Server, use the -e flag, as follows:

```
start_server -e siebel prod01 prod02
```

- To start only Siebel Servers that are marked with the autostart attribute, use the -a flag, as follows:

```
start_server -a
```

Typically, use the -a flag only when invoking the `start_server` script from an autostart script. For more information about the autostart script, see *Siebel Installation Guide* .

- To force the startup, use the -f flag, as follows:

```
start_server -f
```

This flag can be used to make sure that the Siebel Server (or servers) start even if it was not previously shut down cleanly. This flag is typically not needed.

Note: Do not manually copy or rename the svc.siebsrvr.* filenames or any files in the SIEBSRVR_ROOT `/sys` directory. Additionally, do not delete the MW* (mainwin) files stored in the `/tmp` directory while the Siebel Server is running. These files are reused by components in the event of a component restart.

Stopping the Siebel Server System Service on UNIX

This procedure describes how to stop the Siebel Server system service on UNIX.

To stop the Siebel Server system service on UNIX

1. On the Siebel Server computer, log in as the Siebel Service owner user.
2. Run the siebenv.sh or siebenv.csh script in the current shell process.

   For more information about these scripts, see *Siebel Installation Guide* .

3. Run the `ps` command and check whether the application container for the Siebel Server (javaw process daemon) is running. Stop it if necessary.

   For more information, see *Starting and Shutting Down a Siebel CRM Deployment*.
4. Enter the following command:

   ```
   stop_server siebel_server_name
   ```

   You can run this script to stop the system service for a specified Siebel Server (or servers), either within a specified Siebel Enterprise Server or across all of the Siebel Enterprise Servers defined for the current installation. With some of the available command-line flags, you can do the following:

   - To stop multiple Siebel Servers, enter the names of the Siebel Servers (separated by spaces), or enter `all` to stop all of the Siebel Servers configured under the specified SIEBEL_ROOT (or all of the Siebel Servers for a particular Siebel Enterprise Server, if the Siebel Enterprise Server name is specified, as described for the use of the -e flag). Use one of the following commands:

     ```
     stop_server server1 server2...
     stop_server all
     ```

   - To specify the Siebel root directory, use the -r flag, as follows:

     ```
     stop_server -r siebel_root
     ```

     Typically, you do not have to use the -r flag, because the script in the previous step sets the SIEBEL_ROOT environment variable. If the variable is not set, then you must specify the Siebel root directory, using this flag, to indicate the Siebel installation under which the Siebel Server (or servers) is running.
   - To limit the operation to Siebel Servers in a specific Siebel Enterprise Server, use the -e flag, as follows:

     ```
     stop_server -e enterprise server1 server2...
     ```

     You do not have to use the -e flag if the SIEBEL_ENTERPRISE environment variable is set in the siebenv.sh or siebenv.csh script during initial configuration. If the variable is not set, then you must specify the Siebel Enterprise Server name. To stop all of the Siebel Servers for all of the Siebel

Enterprise Servers configured for the SIEBEL_ROOT, do not use the -e flag (you might have to unset the SIEBEL_ENTERPRISE environment variable).

- To stop the Siebel Server system service for the prod01 server in the Siebel Enterprise Server, use the -e flag, as follows:

```
stop_server -e siebel prod01
```

- To force the shutdown, use the -f flag, as follows:

```
stop_server -f
```

This flag causes the Siebel Server to shut down sooner, but it might not give all of the components a chance to shut down cleanly. In general, use the -f flag only if the Siebel Servers did not respond to the unforced shutdown in a timely manner.

Note: A normal shutdown of the Siebel Server system service on AIX, HPUX and SOLARIS deletes MainWin page files of the format MW* stored in the `/tmp` directory. You can safely delete these files if they remain after the Siebel Server shuts down. However, do not delete these files while the Siebel Server is running.

Resetting the Siebel Server System Service on UNIX

This procedure describes how to reset the Siebel Server system service on UNIX.

To reset the Siebel Server system service on UNIX

1. On the Siebel Server computer, log in as the Siebel Service owner user.
2. Run the siebenv.sh or siebenv.csh script to set Siebel environment variables.

   For more information about these scripts, see *Siebel Installation Guide* .
3. Run the `ps` command and check whether the application container for the Siebel Server (javaw process daemon) is running. You can shut it down and, if necessary, restart it later.

   For more information, see *Starting and Shutting Down a Siebel CRM Deployment*.
4. To reset the Siebel Server system service, enter a command like the following:

```
reset_server siebel_server_name
```

Note: Use this script only if the Siebel Server system service is unable to start after an abnormal shutdown or crash of the Siebel Server computer. Do not use it as part of the normal operation of the Siebel Server.

You can run this script to reset the system service for a specified Siebel Server (or servers), either within a specified Siebel Enterprise Server or across all of the Siebel Enterprise Servers defined for the current installation. The names of one or more Siebel Servers (separated by spaces) must be specified on the command line.

The following are explanations of some of the available command-line flags:

- To reset multiple Siebel Servers, enter the names of the Siebel Servers (separated by spaces), or enter `all` to reset all of the Siebel Servers configured under the specified SIEBEL_ROOT (or all of the Siebel Servers

for a particular Siebel Enterprise Server, if the Siebel Enterprise Server name is specified, as described in the use of the -e flag). Use one of the following commands:

```
reset_server server1 server2... reset_server all
```

- To specify the Siebel root directory, use the -r flag, as follows:

```
reset_server -r siebel_root
```

Typically, you do not have to use the -r flag, because the script in the previous step sets the SIEBEL_ROOT environment variable. If the variable is not set, then you must specify the Siebel root directory, using this flag, to indicate the Siebel installation under which the Siebel Server (or servers) is running.

- To specify the Siebel Enterprise Server under which the specified Siebel Server (or servers) is configured, use the -e flag, as follows:

```
reset_server -e enterprise server1 server2...
```

You do not have to use the -e flag if the SIEBEL_ENTERPRISE environment variable is set in the siebenv.sh or siebenv.csh script during initial configuration. If the variable is not set, then you must specify the Siebel Enterprise Server name to indicate the Siebel Enterprise Server under which the Siebel Servers are configured. To reset all of the Siebel Servers for all of the Siebel Enterprise Servers configured for the SIEBEL_ROOT, do not use the -e flag (you might have to unset the SIEBEL_ENTERPRISE environment variable). Instead, use `all` for the enterprise variable.

- To reset the Siebel Server system service for the prod01 server in the Siebel Enterprise Server, use the -e flag in a command like the following:

```
reset_server -e siebel prod01
```

Renaming Servers for an Existing Siebel CRM Deployment

This topic describes the steps to take when one or more host names have changed for servers that host Siebel CRM server modules in an existing deployment of Siebel CRM. Such cases require updates to be made in multiple locations and involve mandatory downtime. Perform the following tasks on all applicable nodes. Review all of the tasks first and confirm the specific requirements for each particular case.

> **Note:** Before you perform these steps, first shut down the Siebel deployment by stopping all of the Siebel services and processes, including the application containers, as described in *Shutting Down a Siebel CRM Deployment* and related topics. Steps for restarting all of the services and processes are included in these procedures.

To perform the necessary steps on each Siebel Application Interface node

1. Back up the `SIEBEL_ROOT\applicationcontainer_external\webapps\applicationinterface.properties` file.
2. Open the applicationinterface.properties file in a text editor like Wordpad or vi.

3. Where the Siebel Gateway that this instance of Siebel Application Interface connects to has moved to a different server or has been renamed, modify the host name for the Siebel Gateway, which is defined in the CGHostURI property.
4. Save the applicationinterface.properties file.

To perform the necessary steps on each Siebel Gateway node

1. Where this Siebel Gateway has moved to a different server or has been renamed, modify the host name in the gateway.properties file, as follows:
    a. Back up the `SIEBEL_ROOT\applicationcontainer_internal\webapps\gateway.properties` file.
    b. Open the gateway.properties file in a text editor like Wordpad or vi.
    c. Modify the Siebel Gateway host name, which is defined in the Gateway, TLSGateway, and Registryhostname properties.
    d. Save the gateway.properties file.
2. If you are using Oracle Enterprise Manager for Siebel CRM, then, where the Siebel Gateway has moved to a different server or has been renamed, modify the host name in the EMdiscovery.properties file, as follows:
    a. Back up the `SIEBEL_ROOT\EMdiscovery.properties` file.
    b. Open the EMdiscovery.properties file in a text editor like Wordpad or vi.
    c. Modify the Siebel Gateway host name defined in the AIConnStr property.
    d. Save the EMdiscovery.properties file.
3. Where any Siebel CRM module has moved to a different server or has been renamed, modify the host name as follows:
    a. Back up the `SIEBEL_ROOT\gtwysrvr\registry\conf\registry.cfg` file.
    b. Back up the `SIEBEL_ROOT\gtwysrvr\registry\conf\version_2` directory.
    c. Open the registry.cfg file in a text editor like Wordpad or vi.
    d. Modify the host names for all applicable Siebel CRM modules.
    e. Save the registry.cfg file.
4. Modify the Gateway Registry service (ZooKeeper), as follows:
    a. Start the Gateway Registry service (ZooKeeper).
    b. Connect to the Gateway Registry using any ZooKeeper client.
    c. Navigate to /ROOT/Config/Profiles/Enterprises.
    d. Select the profile that is used to deploy the Siebel CRM enterprise and modify the host name defined in the NameserverHostName parameter.
    e. Navigate to /ROOT/Config/Profiles/SWSM.
    f. Select the profile that is used to deploy the Siebel Application Interface and modify the host name defined in the GatewayHost parameter. Repeat this step for all the profiles that are used to deploy Application Interface nodes.
    g. Navigate to /ROOT/Config/Profiles/Servers.
    h. Select the profile that is used to deploy the Siebel Server and modify the host name defined in the NameserverHostName parameter. Repeat this step for all the profiles that are used to deploy Siebel Server nodes.
    i. Navigate to every hierarchy up to the leaf level under /ROOT/Config/Deployments. Modify the host name defined in the PhysicalHostIP parameter.

To perform the necessary steps on each Siebel Server node

1. Start the Siebel Gateway Name Server system service.
2. Start Server Manager and connect to the enterprise.
3. Execute the following commands to see the current values for these parameters:
   - `list hidden param DSGatewayAddress for named subsystem ServerDataSrc(`
   - `list param DSConnectString for named subsystem GatewayDataSrc`
   - `list param HOST for server <$ServerName>`

   Repeat this command for all the servers.
4. Execute the following commands to modify the current values for these parameters:
   - `change param DSGatewayAddress=<value> for named subsystem ServerDataSrc` (where the Siebel Gateway has moved to a different server or has been renamed)
   - `change param DSConnectString=<value> for named subsystem GatewayDataSrc` (where the Siebel Gateway has moved to a different server or has been renamed)
   - `change param host=<value> for server <$ServerName>` (where the Siebel Server has moved to a different server or has been renamed)

   Repeat this command for all the servers.
5. Recreate the Siebel Server services, as follows:
   a. Navigate to `<SIEBEL_ROOT>\siebsrvr\bin`.
   b. Execute the following commands:
      - `siebctl -h <$SIEBEL_ROOT> -S siebsrvr -i "<enterprise name>_<Siebel Server name>" -d`
      - `siebctl -h <$SIEBEL_ROOT> -a -f -S siebsrvr -i "<Enterprise name>_<Siebel Server name>" -g <Gateway New Hostname>:<Gateway Port> -e <Enterprise name> -s <Siebel Server name> -l <language> -u "<User Id>" -e <Enterprise name) -A Yes` (where the Siebel Gateway has moved to a different server or has been renamed)

   c. Start all the Siebel Server services.
   d. Start all the Application Interface services.
6. Verify that all modules have started correctly and are functioning normally.

   For more information, see *Siebel Installation Guide* .

6 Administering Siebel Server Run-Time Operations

Administering Siebel Server Run-Time Operations

This chapter describes Siebel Server run-time administration tasks and processes performed by using the Siebel Server Manager GUI. It includes the following topics:

- *Administering Siebel Servers*
- *Administering Siebel Server Component Groups*
- *Administering Siebel Server Components*
- *Administering Component Jobs*
- *Administering Siebel Server Tasks*

Administering Siebel Servers

This topic lists the administration tasks applicable to Siebel Servers. It includes the following information:

- *Starting a Siebel Server*
- *Shutting Down a Siebel Server*

Related Topics

Siebel Enterprise Server Architecture

Administering the Siebel Server System Service

Related Books

Siebel Installation Guide

Siebel System Monitoring and Diagnostics Guide

Starting a Siebel Server

This topic describes how to start a Siebel Server.

Note: In order to start the Siebel Server, the Siebel Server system service must be running.

If the Auto Startup Mode (alias AutoStart) Siebel Server parameter is set to the default value of True, then the Siebel Server starts automatically when the Siebel Server system service is started.

Note: Starting a Siebel Server starts the default number of tasks, as defined in the Default Tasks (alias DfltTasks) parameter, for each background mode component.

This topic is part of *Administering Siebel Servers*.

To start a Siebel Server

1. Navigate to Administration - Server Management screen, then the Enterprises view.
2. In the Servers list, select the Siebel Server of interest.
3. Click Startup.

   The State field changes to Starting Up.

Shutting Down a Siebel Server

This topic describes how to shut down a Siebel Server.

CAUTION: If you shut down the Siebel Server that is hosting your current Siebel Web Client session, then you lose connection to the Siebel Server and cannot restart the Siebel Server by using the Siebel Web Client. In such a case, you must restart the Siebel Server by using the Siebel Developer Web Client or the Server Manager command-line interface. You can shut down and restart Siebel Servers that do not host your Siebel Web Client session without losing your connection.

This topic is part of *Administering Siebel Servers*.

To shut down a Siebel Server

1. Navigate to Administration - Server Management screen, then the Enterprises view.
2. In the Servers list, select the Siebel Server of interest.
3. Click Shutdown.

   The State field changes to Shutting Down.

Note: Shutting down the Siebel Server does not automatically shut down the Siebel Server system service. In some cases, you must shut down not only the Siebel Server, but also the Siebel Server system service.

Administering Siebel Server Component Groups

This topic lists the administration tasks applicable to component groups on a Siebel Server. Enabling or disabling a server component group makes all of the components within the component group available or unavailable.

This topic contains the following information:

- *Enabling a Component Group on a Siebel Server*
- *Disabling a Component Group on a Siebel Server*

Related Topics

Siebel Enterprise Server Architecture

Enabling a Component Group on a Siebel Server

This topic describes how to enable a component group on a Siebel Server.

This topic is part of *Administering Siebel Server Component Groups*.

To enable a component group on a Siebel Server

1. Navigate to Administration - Server Configuration screen, then the Enterprises view.
2. In the Enterprise Servers list, select the enterprise of interest.
3. In the Component Group list, select the component group of interest (that is currently disabled).
4. Click Enable.

   The Enable State field of the component group record changes to Enabled.

Related Topics

Disabling a Component Group on a Siebel Server

Disabling a Component Group on a Siebel Server

This topic describes how to disable a component group on a Siebel Server.

This topic is part of *Administering Siebel Server Component Groups*.

To disable a component group on a Siebel Server

1. Navigate to Administration - Server Configuration screen, then the Enterprises view.
2. In the Enterprise Servers list, select the enterprise of interest.
3. In the Component Group list, select the component group of interest (that is currently enabled).
4. Click Disable.

   The Enable State field of the component group record changes to Disabled.

Related Topics

Enabling a Component Group on a Siebel Server

Administering Siebel Server Components

This topic lists the administration tasks applicable to Siebel Server components. It includes the following information:

- *Starting a Server Component on a Siebel Server*
- *Shutting Down a Server Component on a Siebel Server*

- *Recovering an Unavailable Server Component*
- *Pausing Server Components on a Siebel Server*
- *Resuming Server Components on a Siebel Server*

Related Topics

About Starting Siebel Server Components

Siebel Enterprise Server Architecture

Starting a Server Component on a Siebel Server

This topic describes how to start a server component on a Siebel Server.

This topic is part of *Administering Siebel Server Components*.

To start a server component on a Siebel Server

1. Navigate to Administration - Server Management screen, then the Components view.
2. In the Components list, select the server component of interest.

   Note: Make sure that you select the server component on the Siebel Server of interest.

3. Click Startup.

Shutting Down a Server Component on a Siebel Server

This topic describes how to shut down a server component on a Siebel Server.

This topic is part of *Administering Siebel Server Components*.

To shut down a server component on a Siebel Server

1. Navigate to Administration - Server Management screen, then the Components view.
2. In the Components list, select the server component of interest.

   Note: Make sure that you select the server component on the Siebel Server of interest.

3. Click Shutdown.

Recovering an Unavailable Server Component

This topic describes how to recover a server component that has an Unavailable state.

This topic is part of *Administering Siebel Server Components*.

To recover a server component from an unavailable component state

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

2. In the Components list, select the unavailable server component of interest.
   Note: Make sure that you select the server component on the Siebel Server of interest.
3. Click Shutdown.
   After the server component shuts down, its state changes to Shutdown.
4. After the server component shuts down, click Startup.
   When the server component starts, its state changes to Starting Up. After the server component has started, the state changes to Running.

Pausing Server Components on a Siebel Server

This topic describes how to pause a server component on a Siebel Server.

This topic is part of *Administering Siebel Server Components*.

To pause a server component on a Siebel Server

1. Navigate to Administration - Server Management screen, then the Components view.
2. In the Components list, select the server component of interest.
   Note: Make sure that you select the server component on the Siebel Server of interest.
3. Click Pause.

Resuming Server Components on a Siebel Server

This topic describes how to resume a server component on a Siebel Server that you previously paused.

This topic is part of *Administering Siebel Server Components*.

To resume a server component on a Siebel Server

1. Navigate to Administration - Server Management screen, then the Components view.
2. In the Components list, select the server component of interest.
   Note: Make sure that you select the server component on the Siebel Server of interest.
3. Click Resume.

Administering Component Jobs

This topic lists the administration tasks applicable to component jobs.

This topic contains the following information:

- *About Component Jobs*

- *Starting a Component Job*
- *Starting a Repeating Component Job*
- *Deleting a Component Job or Repeating Component Job*
- *Canceling a Component Job or Repeating Component Job*
- *Holding a Component Job or Repeating Component Job*
- *Resuming a Component Job or Repeating Component Job*
- *Troubleshooting Component Jobs*

About Component Jobs

A component job is a request for one or more Siebel Server tasks to run. A component job is initiated by either the user or the system, and the Siebel Server runs one or more tasks to fulfill the component job. Component jobs can:

- Be scheduled
- Be repeated
- Use component job templates

Make sure that your Siebel Enterprise and Siebel Server configurations allow tasks to run for a server component.

If you are planning to regularly run component jobs with the same parameter values, then configure a component job template, which is a predefined component job.

Only batch mode component tasks are started by running a component job. Background mode components, such as Workflow Monitor Agent (alias WorkMon), are started by the Siebel Server when the component is started. The parameter Default Tasks (alias DfltTasks) defines the number of tasks started for the background mode component. Background mode component tasks can also be started by using the Server Manager command-line interface program.

This topic is part of *Administering Component Jobs*.

Related Topics

Checking Your Siebel Enterprise and Siebel Server Configurations

Configuring Component Job Templates

Using the Siebel Server Manager Command-Line Interface

Troubleshooting Component Jobs

Starting a Component Job

This topic describes how to create and start an individual component job. You can run a component job by using either server components or component job templates. An individual component job is scheduled to run once at a specific time.

This topic is part of *Administering Component Jobs*.

To start a component job

1. Navigate to Administration - Server Management screen, then the Jobs view.
2. In the Jobs list, click New.

The component job status field changes to Creating.

3. In the Component/Job field, click the drop-down list.

   Note: If the Component/Job drop-down list contains no records, then synchronize the Siebel Server components.

4. In the Component/Job drop-down list, select a server component or component job and click OK.

   If you want to use a component job template for your component job, then you must first define the component job template.

5. In the Job Detail view, enter data in other appropriate fields as described in the following table. Click Menu, and then Save Record.

Field	Description
Scheduled Start	The scheduled start date and time of the component job.
Expiration	The date at which the component job is no longer valid.
Requested Server	Set if you want to target a server component on a specific Siebel Server.
Request Key	Set if you want to target a component or repeating component job to a specific instance of the server component identified by the request key. In all other situations, keep this field blank.
Delete Interval	Set with Delete Unit field, this field determines the length of time before the component job is deleted. If it is not updated, then this field defaults to 1.
Delete Unit	Set with Delete Interval field, this field determines the length of time before the component job is deleted. If it is not updated, then this field defaults to Weeks.
Retry on Error	Check this box to retry the component job in case of error.
Sleep Time	This field is available when the Retry on Error check box is True and determines the amount of time before the component job is retried.
Number of Retries	This field is available when the Retry on Error check box is True and determines the number of times the component job is retried.

6. In the Job Parameters list, add or change any component job parameters for the component job and click New.

   a. In the Name field, click the select button.

      The Job Parameters dialog box appears. The parameters that appear in the Job Parameters dialog box vary depending on the server component that you specified in a prior step.

b. Select a parameter in the Component Parameters dialog box, and modify its value.
c. Click Menu, and then Save Record.

7. In the Jobs list, click Submit Job.
   The Status field changes from Creating to Queued.

Related Topics

About Component Jobs

Configuring Component Job Templates

Starting a Repeating Component Job

Synchronizing Components on a Siebel Enterprise Server

Starting a Repeating Component Job

This topic describes how to create and start a repeating component job. You can define repeating component jobs by using either components or component job templates, and you can schedule component jobs to run repeatedly at specific times over specific intervals. In order to run repeating component jobs, both the Server Request Broker (alias SRBroker) and Server Request Processor (alias SRProc) server components must be running. If you have to run repeating jobs at unspecified times or intervals, then run repeating component jobs by using a component job template.

> **Note:** It is not possible to start a business service directly as a repeating component job. To accomplish this task, you first must include the business service in a workflow process. You can then run the workflow process as a repeating component job for the Workflow Process Batch Manager component (alias WfProcBatchMgr).

This topic is part of *Administering Component Jobs*.

To start a repeating component job

1. Create a component job as described in *Starting a Component Job*, but do not start the component job.
2. In the Job Detail view, enter data in the Repeating Info fields as described in the following table. Click Menu, and then Save Record.
3. In the Jobs list, click Submit Job.
   The Status field changes from Creating to Active.

Field	Description
Repeating?	Select this check box to make the component job a repeating component job.
Repeat Unit	This field determines the length of time before the component job repeats when it is set with the Repeat Interval field.
Repeat Interval	Set with Repeat Unit field, this field determines the length of time before the component job repeats.
Repeat From	This field has three possible settings:

Field	Description
	Scheduled Start. Starts the next iteration of the repeating component job after the interval period has elapsed. The interval period is calculated from the time indicated by the Scheduled Start field. For example, if the repeat interval is 5 minutes, the scheduled start time is 09:00, and the component job starts at 09:02, then the next component job is scheduled to start at 09:05. Using this setting might result in a backlog of component jobs that will be started if the actual start time is later than the scheduled start time. **Actual Start.** Starts the next iteration of the repeating component job after the interval period has elapsed. The interval period is calculated from the time the component job starts rather than the time indicated by the Scheduled Start field. For example, if the repeat interval is 5 minutes, the scheduled start time is 09:00, and the component job starts at 09:02, then the next component job is scheduled to start at 09:07. **End.** Starts the next iteration of the repeating component job after the previous component job ends and the repeat interval has elapsed. For example, if the repeat interval is 5 minutes, the scheduled start time is 09:00, the component job starts at 09:02, and the first component job ends at 09:04, then the next component job is scheduled to start at 09:09.
Repetitions	This field determines the number of times the component job repeats.

Related Topics

Configuring Component Job Templates

About Component Jobs

Starting a Component Job

Deleting a Component Job or Repeating Component Job

This topic describes how to delete a component job. You can only delete component jobs that have a status of Creating. After the component job has been submitted, you can only cancel the component job.

> **Note:** You can also delete component jobs by setting the Delete Interval and Delete Units field while creating the component job. After the component job has been started, these fields are read-only.

This topic is part of *Administering Component Jobs*.

To delete a component job or repeating component job during its creation

1. Navigate to Administration - Server Management screen, then the Jobs view.
2. In the Jobs list, select the component job that has a Status field value of Creating.
3. Click Delete.

Completed component jobs are deleted automatically after a configurable period of time. The fields Delete Interval and Delete Units determine this period and are set, by default, to one week.

Related Topics

Starting a Component Job

Administering Component Jobs

Canceling a Component Job or Repeating Component Job

Canceling a Component Job or Repeating Component Job

This topic describes how to cancel component jobs. You can only cancel component jobs that have a state of Queued or On-Hold.

This topic is part of *Administering Component Jobs*.

To cancel a component job or repeating component job

1. Navigate to Administration - Server Management screen, then the Jobs view.
2. In the Jobs list, select the component job that has a Status field value of Queued or On-Hold.
3. Click Cancel Job.

Canceled component jobs are deleted automatically after a configurable period of time. The fields Delete Interval and Delete Units determine this period and are set, by default, to one week.

Related Topics

Starting a Component Job

Holding a Component Job or Repeating Component Job

This topic describes how to hold component jobs. You can only hold component jobs that have a state of Queued.

Holding a component job pauses the applicable task (if the task can be paused) or allows the current task to finish. No new tasks begin for the held component job.

This topic is part of *Administering Component Jobs*.

To hold a component job or repeating component job

1. Navigate to Administration - Server Management screen, then the Jobs view.
2. In the Jobs list, select the component job that has a Status field value of Queued.

   Note: You cannot hold jobs with an Active value in the Status field.

3. Click Hold Job.

Resuming a Component Job or Repeating Component Job

This topic describes how to resume component jobs put on hold. You can only resume component jobs that have a state of On Hold.

This topic is part of *Administering Component Jobs*.

To resume a component job or repeating component job on hold

1. Navigate to Administration - Server Management screen, then the Jobs view.
2. In the Jobs list, select the component job that has a Status field value of On Hold.
3. Click Resume Job.

Related Topics

Holding a Component Job or Repeating Component Job

Troubleshooting Component Jobs

This topic provides guidelines for resolving problems with component jobs.

This topic is part of *Administering Component Jobs*.

To resolve the problem, look for it in the list of symptoms or error messages in the following table.

Problem	Cause	Solution
The Component/Jobs drop-down list contains no records.	Batch mode components are not synchronized.	Synchronize the batch mode components.
	Server component job is based on a new component definition that was added by using the component definition view.	Synchronize the batch mode components.

Related Topics

Synchronizing Components on a Siebel Enterprise Server

Administering Siebel Server Tasks

This topic lists the administration tasks applicable to Siebel Server tasks. It contains the following information:

- *Starting a Siebel Server Task*
- *Pausing a Siebel Server Task*
- *Resuming a Siebel Server Task*
- *Stopping a Siebel Server Task*
- *Terminating a Siebel Server Task*
- *Configuring Siebel Server Task Dynamic Parameters*

Starting a Siebel Server Task

To start a Siebel Server task, you must run a component job. For information about component jobs and administering component jobs, see the following topics:

- *About Component Jobs*
- *Administering Component Jobs*
- *Starting a Component Job*

This topic is part of *Administering Siebel Server Tasks*.

Pausing a Siebel Server Task

This topic describes how to pause a Siebel Server task. You can pause Siebel Server tasks only for certain component types. The following table lists the component types and the predefined components that have this feature.

This topic is part of *Administering Siebel Server Tasks*.

To pause a Siebel Server task

1. Navigate to Administration - Server Management screen, then the Tasks view.
2. In the Tasks list, select the Siebel Server task of interest.
3. Make sure that the Siebel Server task is on the Siebel Server of interest and is of a component type that can be paused. See the following for a list of Siebel Server component types that can be paused.
4. Click Pause.

The following table lists component types that can be paused.

Component Types	Predefined Component	Predefined Component Alias
MailMgr	Email Manager	MailMgr
MktgSrvr	Marketing Server	MktgSrvr
PageMgr	Page Manager	PageMgr
ServerMgr	Server Manager	ServerMgr
TxnMerge	Transaction Merger	TxnMerge
TxnProc	Transaction Processor	TxnProc
TxnRoute	Transaction Router	TxnRoute
WorkActn	Workflow Action Agent	WorkActn

Component Types	Predefined Component	Predefined Component Alias
WorkMon	Workflow Monitor Agent	WorkMon

Related Topics

Administering Siebel Server Tasks

Resuming a Siebel Server Task

Resuming a Siebel Server Task

This topic describes how to resume a paused Siebel Server task.

This topic is part of *Administering Siebel Server Tasks*.

To resume a paused Siebel Server task

1. Navigate to Administration - Server Management screen, then the Tasks view.
2. In the Tasks list, select the paused Siebel Server task of interest.
3. Click Resume.

Related Topics

Administering Siebel Server Tasks

Pausing a Siebel Server Task

Stopping a Siebel Server Task

This topic describes how to stop a Siebel Server task. You can stop running or paused Siebel Server tasks.

> **Note:** It is preferable to stop the individual tasks for a given server component rather than shutting down the server component.

This topic is part of *Administering Siebel Server Tasks*.

To stop a Siebel Server task

1. Navigate to Administration - Server Management screen, then the Tasks view.
2. In the Tasks list, select the running Siebel Server task of interest.
3. Click Stop.

Related Topics

Administering Siebel Server Tasks

Terminating a Siebel Server Task

This topic describes how to terminate a Siebel Server task. Terminating a Siebel Server task signals the Siebel Server to use the operating system control to terminate the task.

This topic is part of *Administering Siebel Server Tasks*.

To terminate a Siebel Server task

1. Navigate to Administration - Server Management screen, then the Tasks view.
2. In the Tasks list, select the running Siebel Server task of interest.
3. Click Stop three times in succession.

Related Topics

Administering Siebel Server Tasks

Configuring Siebel Server Task Dynamic Parameters

This topic describes how to configure dynamic parameters of a Siebel Server task. Siebel Server task parameters consist of Siebel Enterprise Server, Siebel Server, and server component-level parameters, as well as task-specific parameters specified when starting a component job. After a task is running, only dynamic parameters can be changed.

This topic is part of *Administering Siebel Server Tasks*.

To configure Siebel Server task dynamic parameters

1. Navigate to Administration - Server Management screen, then the Tasks view.
2. In the Tasks list, select the running Siebel Server task of interest.
3. Click the Parameters tab.
4. In the Task Parameters list, change the values of the parameters that you want to modify. (Identify dynamic parameters by a check in the Effective Immediately column.)

    a. Select the parameter that you want to modify.
    b. In the Value field, type in the new value.
    c. Click Menu, and then Save Record.

Related Topics

About Siebel System Parameters

About Task Parameters

Administering Siebel Server Tasks

Siebel Enterprise, Server, and Component Parameters

7 Using the Siebel Server Manager Command-Line Interface

Using the Siebel Server Manager Command-Line Interface

This chapter describes how to use the Siebel Server Manager command-line interface. It includes the following topics:

- *Starting the Siebel Server Manager Command-Line Interface*
- *Recommendations for Using the Command-Line Interface*
- *Siebel Server Manager Commands*

Starting the Siebel Server Manager Command-Line Interface

The topics in this chapter describe the procedures available from the Siebel Server Manager command-line interface, which is also known as the srvrmgr program. This program is available on both the Windows and UNIX environments.

An overview of the srvrmgr program and its administration is followed by individual commands used to administer the Siebel Enterprise Server, individual Siebel Servers, and Siebel Server components and component groups. You must have administrative responsibilities defined by the Siebel application and have a user definition in the database in order to access and use the Siebel Server Manager command-line interface.

> **Note:** When using the Siebel Server Manager command-line interface, only use ASCII characters. If you want to enter parameters containing non-ASCII characters (for example, accented French characters, or Russian, Arabic, Japanese, Chinese, Korean, or Thai characters), then use the Siebel Server Manager GUI.

Starting the Siebel Server Manager Program

This procedure describes how to start the Siebel Server Manager program, srvrmgr.

To start the srvrmgr program

1. For Windows servers only: at the DOS prompt, change to the `bin` subdirectory within the Siebel Server root directory:

   ```
   cd SIEBSRVR_ROOT\bin
   ```

   > **Note:** You cannot use the Uniform Naming Convention (UNC) in the Siebel Server Manager command when specifying the path and computer names.

2. Execute the srvrmgr program using flags to specify the parameters that you want:

   ```
   srvrmgr flags
   ```

   For a list of srvrmgr flags, see *Command-Line Flags for the Siebel Server Manager Program*.
3. After the Siebel Server Manager has started, the prompt changes to:

   ```
   srvrmgr: server_name>
   ```

   The server_name parameter appears in the prompt only if you executed the srvrmgr program by specifying a Siebel Server using the /s or -s flag, or after you specify a Siebel Server by using the `set server` command.

   For example, to start the srvrmgr program using the parameters specified in the following table on a Windows server, enter a command like:

   ```
   srvrmgr /g gateway1:2320 /e enterprise1 /s server1 /u sadmin /p pwd
   ```

   To start the srvrmgr program on a UNIX server using the parameters described in the following table, enter a command like the following:

   ```
   srvrmgr -g gateway1:2320 -e enterprise1 -s server1 -u sadmin -p pwd
   ```

The following table shows example parameters for starting the srvrmgr program.

Siebel Gateway	Enterprise	Siebel Server	User Name	Password
gateway1	enterprise1	server1	sadmin	pwd

Command-Line Flags for the Siebel Server Manager Program

The following table lists the command-line flags available for the Siebel Server Manager program, srvrmgr.

Windows Flag	UNIX Flag	Parameter	Description	Required
/b	-b	Not applicable	Batch mode (use with /i to indicate exit when an error is encountered).	No
/c	-c	"command"	Executes a single command (the command must be bounded within double quotes).	No
/e	-e	entrpr_server	Siebel Enterprise Server name.	Yes
/g	-g	gateway_server	Network address of the Siebel Gateway. This value is usually provided with	Yes

Windows Flag	UNIX Flag	Parameter	Description	Required
			the TLS port for the Siebel Gateway, as specified during installation. For example, the value provided using this flag might be gateway1:2320.	
/h or /?	-h or -?	Not applicable	Prints a usage help message	No
/i	-i	input_file	Gets commands from the input file	No
/k	-k	delimiter	Uses delimiter specified to parse columns in output file	No
/l	-l	language	Language code (default is ENU)	No
/m	-m	Not applicable	Compression enabled	No
/o	-o	output_file	Logs information generated in interactive mode to the specified output file. The types of information logged include, for example, the command issued, command output, type of task, task status, start time, and end time. Use this flag with either the flag that specifies a command to log `(c)` or the flag that specifies an input file with numerous commands `(i)`.	No
/p	-p	password	Siebel Server administrator password	Yes, unless /ip is included
/r	-r	Not applicable	Encryption for network packets is enabled (default is N)	No
/s	-s	siebel_server	Siebel Server name (the default is all servers). Starting srvrmgr with this flag connects the program only with that specific Siebel Server. All commands and user authentication are sent only to that Siebel Server. You cannot change the targeted Siebel Server in this mode.	No
/u	-u	username	Siebel Server administrator user name **Note:** The srvrmgr program expects the database to store user names in upper-case format. User names are automatically converted to upper case during the authentication process and login issues result if database user names are stored in lower case.	Yes

Windows Flag	UNIX Flag	Parameter	Description	Required
/z	-z	server_group_name	Server group name. Starting srvrmgr with this flag connects the program to the specified server group and, as a result, to all of the Siebel Servers that are assigned to the server group. For information about managing the assignment of Siebel Servers with server groups, see *Server Group Management Commands*.	No
/ip	-ip	password_file	Provides a file name containing an encrypted version of the Siebel Server administrative password. See additional notes below.	Yes, unless /p is included.

About using an Encrypted Password File

For security reasons, it may not be desirable to include the actual Siebel Server administrator password on the command line. In this scenario, do a one time encryption of the password using the `EncryptString` utility and place the resulting encrypted string into a text file. You can then use the `/ip` or `-ip` command line flag to specify that file.

For example, enter the following command to create the password file:

```
encryptstring mypassword > password_file.txt
```

When launching the Siebel Server Manager, include this file with the `/ip` or `-ip` command, for example:

```
srvrmgr /u SADMIN /p password_file.txt /g gateway /e enterprise
```

About Error Handling for the Siebel Server Manager Program

Error codes are returned by the Siebel Server Manager program (srvrmgr) in the manner described in this topic. Program behavior is described for different usage cases, including those based on the flags that you use to run srvrmgr. Any nonzero values returned are those that might be expected from a shell script. In earlier versions of Siebel CRM, srvrmgr returns 0 (zero) in all of the following cases.

- When you run srvrmgr with valid syntax and valid arguments for flags and no error occurs, then srvrmgr returns 0 (zero).
- When you run srvrmgr with invalid syntax, such as without the required flag /g or -g, then srvrmgr returns a nonzero value.
- When you run srvrmgr with valid syntax but with invalid arguments for any flags, such as with an invalid user name with the /u or -u flag, then srvrmgr returns a nonzero value.
- When you run srvrmgr with the /c or -c flag:
  - If no error occurs, then srvrmgr returns 0 (zero).
  - If any error occurs, then srvrmgr returns a nonzero value.

- When you run srvrmgr with the /i or -i flag and without the /b or -b flag:
  - If no error occurs, then srvrmgr returns 0 (zero).
  - If one or more command process specified in the input file fails, then srvrmgr returns 0 (zero).
  - If srvrmgr cannot connect with the Siebel Gateway or cannot open the specified input or output file, then srvrmgr returns a nonzero value.
- When you run srvrmgr with the /i or -i flag and the /b or -b flag:
  - If no error occurs, then srvrmgr returns 0 (zero).
  - If any error occurs, then srvrmgr returns a nonzero value.

Recommendations for Using the Command-Line Interface

This topic provides recommendations for using the Server Manager command-line interface.

- Target specific Siebel Servers without using the /s or -s flag:
  - Use the directive `for server siebel_server_name` in individual commands. Specifying the name of a specific Siebel Server targets the command to only that Siebel Server.
  - Use a partial name with the % wildcard character to target the command to all of the Siebel Servers with names matching the pattern. Only patterns that start or end with the wildcard character are matched. Wildcards in the middle of the string are not matched. For example, the following command lists components for all of the Siebel Servers with a name beginning with WF:

    ```
    list components for server WF%
    ```

  - Use the command `set server siebel_server_name` . To return to the mode where commands are targeted to all of the Siebel Servers, use `unset server`. For more information about these commands, see *Siebel Server Manager Environment Commands*.

    > **Note:** When you are using the `set` command, the connections to other Siebel Servers are maintained and continue to run.

- Start srvrmgr using the /s or -s flag for frequent list operations. Parse the resulting data for each Siebel Server. Aggregate the list data for the enterprise externally to the srvrmgr process. This method improves performance by keeping srvrmgr from serializing the operations.
- Use the `show` clause to specify only the columns with data that you are using. For more information about using the `show` clause, see *List Command Configuration* .
- Use the /i or -i option to open a single long-running srvrmgr session and send it commands rather than using the /c or -c option. You can also execute commands conditionally from a script, by using the /i or -i option.
- When using srvrmgr commands from a file or script, use the `sleep` command to configure wait periods (in seconds) before the next srvrmgr command. For example, after starting the Siebel Server, use the `sleep` command to wait until the Siebel Server and its component are running before issuing the next command.
- Use the `read` command during an active srvrmgr session to dynamically input srvrmgr commands from a file.
- Specify a value for the parameter TaskTag when starting a new task. This text appears in the `list tasks` command if you include the TK_TASKTAG column. For example, enter a command like the following:

```
list tasks show TK_TASKTAG
```

- Start srvrmgr using the /z or -z flag to connect to a server group. For example, on a Windows server, enter a command like the following:

  ```
  srvrmgr /g gateway1 /e enterprise1 /z server_group_name /u sadmin /p pwd
  ```

  On a UNIX server, enter a command like the following:

  ```
  srvrmgr -g gateway1 -e enterprise1 -z server_group_name -u sadmin -p pwd
  ```

  This command connects you to all of the Siebel Servers assigned to the server group.

Siebel Server Manager Commands

After the Siebel Server Manager has been started, you can execute administrative tasks by using the commands described in this topic.

These commands can also be written into an ASCII text file, exactly as you execute them through the Siebel Server Manager, and used as a batch input file by running srvrmgr by using the /i or -i flag. Running the batch input file is especially useful in the administration of similar Siebel Server component definitions for multiple Siebel Servers.

Note: You must have the Siebel Administrator responsibility to connect with the srvrmgr command-line tool.

The Siebel Server Manager commands are divided into the following categories:

- **Help.** For more information, see *Help Commands* .
- **Environment.** For more information, see *Siebel Server Manager Environment Commands*.
- **List.** For more information, see *List Commands* and *List Command Configuration* .
- **Server group management.** For more information, see *Server Group Management Commands*.
- **Siebel Server management.** For more information, see *Siebel Server Management Commands*.
- **Component group definition.** For more information, see *Component Group Definition Commands*.
- **Component definition.** For more information, see *Component Definition Commands* and *Reconfiguring Component Definition Commands*.
- **Component management.** For more information, see *Component Management Commands*.
- **Task management.** For more information, see *Task Management Commands*.
- **Parameter management.** For more information, see *Parameter Management Commands*.
- **Named subsystem management.** For more information, see *Named Subsystem Management Commands*.
- **System alert notification.** For more information, see *System Alert Notification Commands*.
- **List definition.** For more information, see *List Definition Commands* and *List Parameter Override Commands*.
- **Event logging.** For more information, see *Event Logging Commands*.
- **Preferences.** For more information, see *Server Manager Command-Line Preferences*.

Command Syntax

This topic lists the command-line syntax and usage for Siebel Server Manager commands.

Component names and parameter names used in the command-line interface differ from the Siebel Server Manager GUI. To get the actual component and parameter names used in the command-line interface, use the list commands. For information about using the list commands, see *List Commands*.

For user-defined values such as siebel_server_name, component_alias_name, and parameter_alias_name, you must enclose these values in quotes if the value:

- Contains spaces
- Is a keyword such as server or component that you do not want to be parsed

For example, you must enclose the Siebel Server name in double quotes for the following command, because the Siebel Server name contains a space:

```
start task for component EIM server "North America" with Config=default.ifb
```

Note: If a srvrmgr command contains nested quotes, that is, quotes contained within quotes, then precede the inner quotes by the backslash escape character (\).

Help Commands

Use the Help command for Siebel Server Manager to retrieve a list of commands or obtain help on a specific command.

To obtain help

- Enter the following command:

  ```
  help
  ```

- For help for a specific command, enter a command like the following:

  ```
  help command
  ```

Siebel Server Manager Environment Commands

Use environment commands to set the Siebel Server Manager environment variables, which control the current Siebel Server Manager session.

To set the current working Siebel Server

- Enter the following command:

  ```
  set server siebel_server_name
  ```

This command works only if you did not specify a Siebel Server when executing the srvrmgr program using the -s flag.

To unset (clear) the current working Siebel Server

- Enter the following command:

```
unset server
```

This command works only if you did not specify a Siebel Server when executing the srvrmgr program using the -s flag.

To show the environment variables

- Enter the following command:

```
show
```

To show an individual environment variable

- Enter the following command:

```
show variable_name
```

To spool output to a file

- Enter the following command:

```
spool output_file
```

To stop spooling to a file

- Enter the following command:

```
spool off
```

To read commands from a file

- Enter the following command:

```
read input_file
```

To refresh the Siebel Enterprise Server connections

- Enter the following command:

```
refresh enterprise
```

The `refresh enterprise` command closes all of the connections to the existing Siebel Servers and creates new connections to these servers.

To remove header and footer information from srvrmgr command-line output

- Enter the following command:

  ```
  set header false
  ```

  and

  ```
  set footer false
  ```

  Removing the header and footer information is useful if you are trying to parse the output of srvrmgr commands.

To add header and footer information to the srvrmgr command-line output

- Enter the following command:

  ```
  set header true
  ```

  and

  ```
  set footer true
  ```

To exit the srvrmgr program

- Enter the following command:

  ```
  exit
  ```

  To save any configuration changes prior to exiting, use a `backup` command listed in *Siebel Server Management Commands*.

List Commands

Use the `list` command to display current data only. This command does not change any data.

To list available Siebel Servers

- Enter the following command:

  ```
  list servers
  ```

- For a component, enter the following command:

  ```
  list servers for component component_alias_name
  ```

- For a component group, enter the following command:

```
list servers for component group component_group_alias_name
```

To list component groups

- For all component groups, enter the following command:

```
list component groups
```

- For a particular Siebel Server, enter the following command:

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

If you are connected to the Siebel Server, then this `list` command list only component groups from shared memory. Otherwise, it lists the component groups assigned to that Siebel Server from the Siebel Gateway. See also the `describe` command in *Component Group Definition Commands*.

To list current component group status

- For all instances of the component group, enter the following command:

```
list component group component_group_alias_name
```

- For a particular Siebel Server, enter the following command:

```
list component group component_group_alias_name for server siebel_server_name
```

To list current component status

- For all components, enter the following command:

```
list component
```

- For all instances of the component, enter the following command:

```
list component component_alias_name
```

- For a particular Siebel Server, enter the following command:

```
list component for server siebel_server_name
```

- For a particular task, enter the following command:

```
list component for task task_number
```

To list values for a particular task, you first must set the current working Siebel Server by using the `set server` command. For information about this command, see *Siebel Server Manager Environment Commands*.

To list subsystems

- For all subsystems, enter the following command:

```
list subsystem
```

To list named subsystems

- For all named subsystems, enter the following command:

```
list named subsystem
```

- For a particular subsystem, enter the following command:

```
list named subsystem for subsystem subsystem_alias_name
```

- For a particular Siebel Server, enter the following command:

```
list named subsystem for server siebel_server_name
```

To list current processes

- To list the status of current tasks, enter the following command:

```
list processes
```

- For all tasks, enter the following command:

```
list tasks
```

- For a particular Siebel Server, enter the following command:

```
list tasks for server siebel_server_name
```

- For a particular component, enter the following command:

```
list tasks for component component_alias_name
```

- For a particular component group, enter the following command:

```
list tasks for component group component_group_alias_name
```

- For a particular task, enter the following command:

```
list task task_number
```

To list values for a particular task, you first must set the current working Siebel Server by using the `set server` command. For information about this command, see *Siebel Server Manager Environment Commands*.

> **Note:** The number of tasks returned is determined by the Maximum Tasks parameter for that component. For more information about the Maximum Tasks parameter, see *Siebel Enterprise, Server, and Component Parameters*.

To list tasks for session mode components

- For a particular Siebel Server, enter the following command:

```
list sessions for server siebel_server_name
```

- For a particular component, enter the following command:

```
list sessions for comp component_alias_name
```

- For a particular Application Object Manager login, enter the following command:

```
list sessions for login object_manager_login
```

- For a list of hung tasks, enter the following command:

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

- For a list of active tasks, enter the following command:

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

To list current parameter values

- For the Siebel Enterprise Server, enter the following command:

```
list ent param
```

- For all Siebel Servers, enter the following command:

```
list parameters
```

- For a particular Siebel Server, enter the following command:

```
list parameters for server siebel_server_name
```

- For a particular component on all Siebel Servers, enter the following command:

```
list parameters for component component_alias_name
```

- For a particular component on a particular Siebel Server, enter the following command:

  ```
  list parameters for component component_alias_name server siebel_server_name
  ```

- For a particular task, enter the following command:

  ```
  list parameters for task task_number server siebel_server_name
  ```

To list current advanced parameter values

- Use the previously documented commands for listing parameters, but preface `advanced` before `parameters`. For example:

  ```
  list advanced parameters for server siebel_server_name
  ```

To list current state values

- For all state values, enter the following command:

  ```
  list state values
  ```

- For a particular Siebel Server, enter the following command:

  ```
  list state values for server siebel_server_name
  ```

- For a particular task, enter the following command:

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

  To list values for a particular task, you first must set the current working Siebel Server by using the `set server` command. For information about this command, see *Siebel Server Manager Environment Commands*.

To list current statistic values

- For all statistics, enter the following command:

  ```
  list statistics
  ```

- For a particular Siebel Server, enter the following command:

  ```
  list statistics for server siebel_server_name
  ```

- For a particular component, enter the following command:

  ```
  list statistics for component component_alias_name
  ```

- For a particular task, enter the following command:

```
list statistics for task task_number
```

  To list values for a particular task, you first must set the current working Siebel Server by using the `set server` command. For information about this command, see *Siebel Server Manager Environment Commands*.

List Command Configuration

The following commands modify or configure the output for the list commands described in *List Commands*.

To modify the output of an individual list command

- To display specific columns, enter the following command:

```
list list_object
show column_1, column_2, ..., column_n
```

  For example:

```
list components show SV_NAME, CC_ALIAS
```

- To display specific columns with a `for` clause, enter the following command:

```
list list_object
 for for_object
 show column_1, column_2, ..., column_n
```

  For example:

```
list components for SRVR_1 show CC_ALIAS
```

To list available columns for a list command

- Enter the following command:

```
configure list list_object
```

To configure the output of the list command

- To display only specific columns, enter the following command:

```
configure list list_object
 show column_1, column_2, ..., column_n
```

  This command changes future `list list_object` commands to display only those columns defined.

> **Note:** Once you configure a specific list command for a given srvrmgr session, it cannot be configured again in that session. A new session must be started to view other columns for that list command.

Server Group Management Commands

Use the server group management commands to manage the assignment of Siebel Servers with server groups. A Siebel Server can only be assigned to one server group at a time. A server group can contain many Siebel Servers.

Once you assign Siebel Servers to a server group, you can specify the server group name as a parameter for the /z or -z flag when starting the srvrmgr program. Specifying the server group connects the srvrmgr program to all of the Siebel Servers assigned to the specified server group.

For example, on a Windows server, enter a command like the following:

```
srvrmgr /g gateway1 /e enterprise1 /z server_group_name /u sadmin /p pwd
```

On a UNIX server, enter a command like the following:

```
srvrmgr -g gateway1 -e enterprise1 -z server_group_name -u sadmin -p pwd
```

For more information about starting the srvrmgr program, see *Starting the Siebel Server Manager Command-Line Interface* .

To assign a Siebel Server to a server group

- Enter the following command:

```
change attribute groupname=server_group_name for server siebel_server_name
```

To unassign a Siebel Server from a server group

- Enter the following command:

```
change attribute groupname=" " for server siebel_server_name
```

> **Note:** Make sure to include a space between the quotation marks.

Siebel Server Management Commands

Use the Siebel Server management commands to start or stop a Siebel Server.

To start a Siebel Server

- Enter the following command:

```
startup appserver siebel_server_name
```

To shut down a Siebel Server

- Enter the following command:

```
shutdown appserver siebel_server_name
```

Component Group Definition Commands

Use these commands to create, delete, assign, remove, enable, and disable component groups.

To create a component group

- Enter the following command:

```
create component group component_group_alias_name full name "descriptive_name"
description "description_of_component_group"
```

To assign a component group to a Siebel Server

- Enter the following command:

```
assign component group component_group_alias_name to server siebel_server_name
```

To unassign a component group from a Siebel Server

- Enter the following command:

```
unassign component group component_group_alias_name from server siebel_server_name
```

> **CAUTION:** Unassigning a component group from a Siebel Server results in a loss of component group customization, for example, parameter settings. Before unassigning a component group, review *About Assigned and Unassigned Component Group*.

To enable a component group for the Siebel Enterprise Server

1. Enter the following command:

```
enable component group component_group_alias_name
```

2. Stop and restart the system service to make the changes take effect.

   For more information about how to stop or start the Siebel Server system service, see *Administering the Siebel Server System Service*.

   This procedure works only if you did not run Siebel Server Manager command-line interface using the /s or -s flag.

   > **Note:** Before you enable a component group for the Siebel Enterprise Server, at least one component in the group must be active.

To enable a component group on a Siebel Server

1. Enter the following command:

   ```
   enable component group component_group_alias_name to server siebel_server_name
   ```

2. Stop and restart the system service to make the changes take effect.

   For more information about how to stop or start the Siebel Server system service, see *Administering the Siebel Server System Service*.

   > **Note:** Use this command when enabling a component that was previously disabled on a particular server. Newly created component groups are enabled by default.

To disable a component group for the Siebel Enterprise Server

1. Enter the following command:

   ```
   disable component group component_group_alias_name
   ```

2. Stop and restart the system service to make the changes take effect.

   For more information about how to stop or start the Siebel Server system service, see *Administering the Siebel Server System Service*.

To disable a component group for a Siebel Server

1. Enter the following command:

   ```
   disable component group component_group_alias_name for server siebel_server_name
   ```

2. Stop and restart the system service to make the changes take effect.

   For more information about how to stop or start the Siebel Server system service, see *Administering the Siebel Server System Service*.

To list component groups from the Siebel Gateway

- Enter the following command:

```
describe component group
```

The `describe` command lists the component groups from the Siebel Gateway.

To remove a component group from a Siebel Server

- Enter the following command:

```
remove component group component_group_alias_name from server siebel_server_name
```

To delete a component group

- Enter the following command:

```
delete component group component_group_alias_name
```

In order for you to delete a component group, the component group cannot contain any server components or component definitions.

Component Definition Commands

Use the component definition commands to create, activate, or delete defined components. Component definitions are contained in component groups, both of which are defined at the Siebel Enterprise Server level. To use a new component, make sure that the component definition is activated and the component group containing the new component is assigned to the appropriate server. For component group commands, see *Component Group Definition Commands*.

Note: When working with component definition commands, start and run the srvrmgr program for the enterprise. That is, do not start srvrmgr with the /s or -s flag and do not run the command `set server`.

To create a new component

- Enter the following command:

```
create component definition component_alias_name for component type existing_component_type_alias_name
component group existing_component_group_alias_name
run mode run_mode
full name "component_full_name"
description "description_of_component"
with parameter parameter_alias_name=value
fixparam fixed_parameter_alias_name=fixed_value
```

The run mode options are:

- Batch
- Interactive
- Background

The component alias must be unique across the enterprise. The length of the alias name is dependent on the operating system.

You must enclose in quotes any keywords that you use in the component description, such as the keywords *for* or *component*. The alias or short name is required for the component group that you specify for this component definition. For a list of existing component groups and their corresponding aliases, see *Siebel Server Components*. For more information about component types, see *About Server Component Types*.

After running the `create` command, use the `activate component definition` command to enable the component definition at the enterprise, component definition level, and to enable and assign the component to the component group that you created. This action only occurs if the component definition is in the creating state. If the component definition is not in the creating state, then the command only enables the component definition at the enterprise level.

To activate a component definition

- After defining the component, activate the defined component by entering:

```
activate component definition component_alias_name
```

> **Note:** If you receive an error when attempting to activate a new component definition, then make sure that you did not start the srvrmgr command-line interface program by using the /s or -s flag, which targets only a specific server.

To deactivate a component definition

- Enter the following command:

```
deactivate component definition component_alias_name
```

To delete a component definition

- Enter the following command:

```
delete component definition component_alias_name
```

To copy a Siebel Server component definition

- Enter the following command:

```
copy compdef from source_comp_def_alias to target_comp_def_alias
```

Reconfiguring Component Definition Commands

To reconfigure component definitions, you must start the component reconfiguration, make the necessary configurations (for parameter configuration, see *Parameter Management Commands*), and then commit the reconfiguration. See the following procedures for these commands.

CAUTION: Review the background information about component definition reconfiguration before performing this task. For more information, see *Reconfiguring Siebel Server Component Definitions*.

To start a component definition reconfiguration

- Enter the following command:

```
reconfig compdef component_alias_name
```

To commit a component definition reconfiguration

- Enter the following command:

```
commit reconfig compdef component_alias_name
```

To cancel a component definition reconfiguration

- Enter the following command:

```
cancel reconfig compdef component_alias_name
```

Component Management Commands

Use component management commands to start or shut down Siebel Server components.

The `startup systemcomps`, `shutdown systemcomps`, `startup systemauxcomps`, `shutdown systemauxcomps`, `startup nonsystemcomps`, and `shutdown nonsystemcomps` commands manage the components in the System Management (alias System) component group or the Auxiliary System Management (alias SystemAux) component group, as follows:

- The components in the System component group include Server Manager, Server Request Broker, and Siebel Connection Broker.
- The components in the SystemAux component group include File System Manager, Server Request Processor, Server Tables Cleanup, and Siebel Administrator Notification Component.

To start a Siebel Server component

- Enter the following command:

```
startup component component_alias_name for server siebel_server_name
```

To shut down a Siebel Server component

- Enter the following command:

```
shutdown component component_alias_name for server siebel_server_name
```

To configure a Siebel Server component to start automatically

- Enter the following command:

```
auto start comp component_alias_name for server siebel_server_name
```

To start a Siebel Server component manually

- Enter the following command:

```
manual start comp component_alias_name for server siebel_server_name
```

To start Siebel Server components for the System component group

- Enter the following command:

```
startup systemcomps for server siebel_server_name
```

To shut down Siebel Server components for the System component group

- Enter the following command:

```
shutdown systemcomps for server siebel_server_name
```

To start Siebel Server components for the SystemAux component group

- Enter the following command:

```
startup systemauxcomps for server siebel_server_name
```

To shut down Siebel Server components for the SystemAux component group

- Enter the following command:

```
shutdown systemauxcomps for server siebel_server_name
```

To start Siebel Server components for all components except those in the System or SystemAux component group

- Enter the following command:

```
startup nonsystemcomps for server siebel_server_name
```

To shut down Siebel Server components for all components except those in the System or SystemAux component group

- Enter the following command:

```
shutdown nonsystemcomps for server siebel_server_name
```

Task Management Commands

Use task management commands to manage tasks for components running in batch or background mode.

You can start a new process by using the `start task` command or the `run task` command. Use the `start task` command if you plan to start multiple processes and use the `run task` command if you want to make sure that a process has run to completion.

The following are the two task management commands:

- **Start task.** The `start task` command starts a new process and allows you to execute a new command immediately. You are not notified of the task status, nor are you alerted if the task fails to perform. Instead, use the `list task` command to check the status of processes that were started by using the `start task` command.
- **Run task.** The `run task` command starts a new process that runs to completion (or exits with an error). You cannot execute a new command until the process has run to completion. The task status is displayed as the process is running.

  To use multiple task parameters in a task command, list the parameters in a comma-separated list. The following example shows how to start a new process using various values for a given parameter:

```
start {task | server} for component component_alias_name with parameter_alias_
name=value1, value2, value3
```

To start a new task in batch mode

- Enter the following command:

```
start task for component component_alias_name server siebel_server_name with
parameter_alias_name1=value1, parameter_alias_name2=value2
```

This command starts a new task in batch mode and returns to the Siebel Server Manager immediately.

To start a new task in background mode

- Enter the following command:

```
start server for component component_alias_name server siebel_server_name with
parameter_alias_name1=value1, parameter_alias_name2=value2
```

This command starts a new task in background mode and returns to the Siebel Server Manager immediately.

To run a new task in batch mode

- Enter the following command:

```
run task for component component_alias_name server siebel_server_name with
parameter_alias_name1=value1, parameter_alias_name2=value2
```

This command runs a new task in batch mode to completion before returning to the Siebel Server Manager.

To pause a running task

- Enter the following command:

```
pause task task_ID for server siebel_server_name
```

> **Note:** Only tasks from certain component types can be paused. For a list of these component types, see *Pausing a Siebel Server Task*.

To resume a paused task

- Enter the following command:

```
resume task task_ID for server siebel_server_name
```

To stop a running task

- Enter the following command:

```
stop task task_ID for server siebel_server_name
```

To terminate a running task using the kill command

- Enter the following command:

```
kill task task_ID for server siebel_server_name
```

The `kill task` command signals the Siebel Server to use operating system control to terminate the task. This command replicates the GUI procedure of clicking Menu and then Stop Task three times in succession on a running task.

Parameter Management Commands

Use the parameter management commands to change the values of a parameter.

To change an enterprise parameter

- Enter the following command:

```
change ent param parameter_alias_name1=value1, parameter_alias_name2=value2
```

To change a component definition parameter

- Enter the following command:

```
change parameter parameter_alias_name1=value1, parameter_alias_name2=value2 for
 compdef component_definition_name
```

To change a component type parameter

- Enter the following command:

```
change parameter parameter_alias_name1=value1, parameter_alias_name2=value2 for
 comptype component_type_name for server siebel_server_name
```

To change a Siebel Server parameter

- Enter the following command:

```
change parameter parameter_alias_name1=value1, parameter_alias_name2=value2 for
 server siebel_server_name
```

To change a component parameter

- Enter the following command:

```
change parameter parameter_alias_name1=value1, parameter_alias_name2=value2 for
component component_alias_name server siebel_server_name
```

> **Note:** If you started srvrmgr with the /s or -s flag, then you do not have to include `server siebel_server_name` for this command.

To change a task parameter

- Enter the following command:

```
change parameter parameter_alias_name1=value1, parameter_alias_name2=value2 for
task task_number
```

> **Note:** After a server, component, or named subsystem parameter is modified, it ignores future parameter changes at higher levels. That is, future parameter changes at higher levels in the hierarchy do not cascade down to lower levels. Use the following commands to delete overrides and reinstate this functionality.

To delete an enterprise parameter override

- Enter the following command:

```
delete enterprise parameter override param parameter_alias_name
```

To delete a Siebel Server parameter override

- Enter the following command:

```
delete parameter override for server siebel_server_name param "parameter_alias_name"
```

To delete a named subsystem parameter override

- Enter the following command:

```
delete parameter override for named subsystem named_subsystem_alias_name param "parameter_alias_name"
```

To delete a server component parameter override

- Enter the following command:

```
delete parameter override for comp component_alias_name server siebel_server_name param
 "parameter_alias_name"
```

To delete a server component definition parameter override

- Enter the following command:

```
delete parameter override for compdef component_alias_name param "parameter_alias_name"
```

Named Subsystem Management Commands

Use named subsystem management commands to create, delete, and modify named subsystems. For more information about named subsystems, see *About Named Subsystem Parameters* and *Application Object Manager Named Subsystem Parameters*.

To create a new named subsystem

- Enter the following command:

```
create named subsystem named_subsystem_alias_name for subsystem
subsytem_alias_name full name named_subsystem_full_name description "My
description" with parameter_alias_name1=value1, parameter_alias_name2=value2
```

For example:

```
create named subsystem MyJMSTestSubsys3 for subsystem JMSSubsys full name
MyJMSTestSubsysNewTest description "Another named subsystem" with
ConnectionFactory="weblogic.examples.jms.QueueConnectionFactory"
```

To delete a named subsystem

- Enter the following command:

```
delete named subsystem named_subsystem_alias_name
```

To list all named subsystem parameters

- For a particular named subsystem, enter the following command:

```
list parameters for named subsystem named_subsystem_alias_name
```

To list a particular named subsystem parameter

- Enter the following command:

```
list parameter parameter_alias_name for named subsystem named_subsystem_alias_name
```

To modify one or more named subsystem parameters

- Enter the following command:

```
change parameter parameter_alias_name1=value1, parameter_alias_name2=value2 for named
 subsystem named_subsystem_alias_name
```

System Alert Notification Commands

Use the following commands to configure system alert notification for server components. For more information about system alert notification, see *About System Alert Notification*.

To troubleshoot any problems with system alert notification, see *Troubleshooting System Alert Notification*.

To set the administrator email address

- Enter the following command:

```
change param AdminEmailAddress=Admin_Email_Address for named subsystem AdminEmailAlert
```

  where Admin_Email_Address is the email address that receives the alert notification email.

To set the SMTP host and port number used for email notifications

- Enter the following command:

```
change param SMTPServer=SMTP_Server for named subsystem AdminEmailAlert
```

  where SMTP_Server is the email server that routes the alert notification email.

To set the From email address

- Enter the following command:

```
change param FromAddress=Server_Email_Address for named subsystem AdminEmailAlert
```

  where Server_Email_Address is the email address that sends the alert notification email.

To test the system alert notification

- Enter the following command:

```
start task for comp AdminNotify server Siebel_Server_Name
```

  where Siebel_Server_Name is the name of the Siebel Server that hosts the AdminNotify server component.

List Definition Commands

Use list definition commands to list definitions for components, parameters, state values, and statistics.

To list component definitions

- For a particular component, enter the following command:

```
list component definitions for component component_alias_name
```

- For a particular task, enter the following command:

```
list component definitions for task task_number
```

List Parameter Override Commands

Use list parameter override commands to list parameter overrides for an enterprise, a Siebel Server, a component, a component definition, or a named subsystem.

For more information about parameter overrides, see *About the Siebel Enterprise Server*. See also *Deleting System Parameter Overrides*.

To list parameters for the enterprise

- Enter the following command:

```
list entparam overrides
```

To list parameters for a Siebel Server

- Enter the following command:

```
list param overrides for server siebel_server_name
```

To list parameters for a component

- Enter the following command:

```
list param overrides for comp component_alias_name server siebel_server_name
```

To list parameters for a component definition

- Enter the following command:

```
list param overrides for compdef component_definition_alias_name
```

To list parameters for a named subsystem

- Enter the following command:

```
list param overrides for named subsystem named_subsystem_alias_name
```

To list parameters for component definitions

- Enter the following command:

```
list param overrides for component component_alias_name
```

Event Logging Commands

Use the event logging commands to list event types for components and to change the values for event log levels. For more information about event logging, see *Siebel System Monitoring and Diagnostics Guide* .

To list event types

- Enter the following command:

```
list evtloglvl for component component_alias_name
```

To change the event log level for a component

- Enter the following command:

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

To change the event log level for a component on a Siebel Server

- Enter the following command:

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

To change the event log level for a Siebel Server

- Enter the following command:

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

Server Manager Command-Line Preferences

You can create aliases for commands and configure list commands to return specific columns. These can be saved in a preferences file which is available to load the next time that you open a Siebel Server manager session. The preferences file is stored in the same directory as the Server Manager program. For the location of the Server Manager program, see *Starting the Siebel Server Manager Command-Line Interface* .

To create an alias for a command

- Enter the following command:
  ```
  alias alias command_name
  ```

  For example, the following command creates an alias `lc` for the command `list components`:
  ```
  srvrmgr> alias lc list components
  ```

To delete an alias for a command

- Enter the following command:
  ```
  unalias alias
  ```

To list the columns returned for a list command

- Enter the following command:
  ```
  configure list_command
  ```

To configure a list command to show specific columns

- Enter the following command:
  ```
  configure list_command show column1, column2, column3...
  ```

  For example, the following command configures the `list components` command to return the component name column only.
  ```
  srvrmgr> configure list components show CC_NAME
  ```

To configure a list command to show all columns

- Enter the following command:
  ```
  configure list_command show all
  ```

  For example, the following command configures the `list components` command to return all columns.
  ```
  srvrmgr> configure list components show all
  ```

  > **Note:** The columns that are returned might not contain useful data. However, by using subsequent commands, you can specify which columns to display.

To save preferences

- Enter the following command:
  ```
  save preferences
  ```

  Preferences are saved in the same directory as the Server Manager program.

To load preferences

- Enter the following command:
  ```
  load preferences
  ```

8 Siebel Server Infrastructure Administration

Siebel Server Infrastructure Administration

This chapter describes how to administer the Siebel Server infrastructure and system management components. It includes the following topics:

- *About Server Request Broker (SRBroker)*
- *Configuring Tasks for Server Request Broker*
- *About Server Request Processor (SRProc)*
- *About Siebel Connection Broker (SCBroker)*
- *About Other System Management Components*
- *Backing Up and Restoring the Siebel Gateway Registry*
- *Administering the Siebel File System*
- *Rolling Back the Siebel Runtime Repository to a Prior Version*

About Server Request Broker (SRBroker)

Server Request Broker (alias SRBroker) is an interactive ode Siebel Server component that belongs in the System Management component group. By default, one SRBroker is started for each Siebel Server. SRBroker handles client component requests by acting as a request router. For example, if a client makes a request to a Siebel Server for a component that is not running on that Siebel Server, then the request is routed to another Siebel Server that is running the requested component.

Siebel Server requests from clients that have no end point get stored in the database until the request is completed. SRBroker works with the Server Request Processor (alias SRProc). For more information about this component, see *About Server Request Processor (SRProc)*.

SRBroker also controls how many component requests by clients can be serviced by a Siebel Server at one time. Each client connection and component connection counts as one task. The number of tasks that can be handled by a single SRBroker is determined by the Maximum Tasks (alias MaxTasks), Maximum MT Servers (alias MaxMTServers), and Minimum MT Servers (alias MinMTServers) component parameters. Keep MaxMTServers and MinMTServers at their default value of 1 for SRBroker. For more information about these parameters, see *Siebel Performance Tuning Guide* . For information about how to set the number of tasks for SRBroker, see *Configuring Tasks for Server Request Broker*.

Do not configure run-time parameters for SRBroker. If you have to support more client and component connections, then increase the number of tasks that can be handled by the SRBroker component.

Configuring Tasks for Server Request Broker

This topic describes how to configure the number of tasks for the Server Request Broker (alias SRBroker) component. For more information about SRBroker, see *About Server Request Broker (SRBroker)*.

To change the number of tasks that can be handled by Server Request Broker

1. Navigate to the Administration - Server Configuration, and 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, query for Server Request Broker (alias SRBroker) in the Component field.
5. Select the Parameters view tab.
6. In the Parameters list, query for the Maximum Tasks (alias MaxTasks) parameter.
7. In the Value on Restart field, type in the number of tasks.

   The default value is 100. For more information about this parameter, see the parameter definition in *Generic Parameters*. For more information about values to set this parameter, see *Siebel Performance Tuning Guide* .
8. For changes to take effect, restart the Siebel Server system service.

   For more information about restarting the Siebel Server system service, see *Administering the Siebel Server System Service*.

About Server Request Processor (SRProc)

The Server Request Processor (alias SRProc) and the Server Request Broker (alias SRBroker) components are jointly responsible for the processing of both synchronous and asynchronous requests from a variety of Siebel Server components. SRProc is a background mode component that handles requests between the Siebel Server and the database. There can only be one instance of SRProc for each Siebel Server. The following components rely on a functioning SRProc and SRBroker:

- Assignment Manager
- Communications Manager
- Enterprise Application Integration
- EIM
- Field Service (all components)
- Interactive Assignment
- Workflow Management

If either SRBroker or SRProc become unavailable for any reason, then the ability to execute intercomponent requests is severely affected. The request mechanism (component jobs) of the Server Manager GUI relies on a functioning SRBroker and SRProc to schedule and process requests. However, the server manager command-line interface program bypasses this request mechanism permitting the user to start (but not schedule) a component task by using

the command-line interface if either or both the SRBroker or SRProc components are unavailable (or, alternatively, restarting the SRBroker or SRProc components). For more information about using the server manager command-line interface program, see *Using the Siebel Server Manager Command-Line Interface*.

Several parameters are available that ensure that these components automatically restart in the event of a failure, so the components experience as little downtime as possible. For information about the parameters Default Tasks (alias DfltTasks) and Auto-Restart (alias AutoRestart), see *Siebel Server Components and Parameters*.

About Siebel Connection Broker (SCBroker)

The Siebel Connection Broker (alias SCBroker) component is a background mode server component that provides intraserver load balancing. By default, it is always enabled and online. At least one instance of SCBroker must be running on any Siebel Server hosting interactive components.

> **Note:** If a Siebel Server hosts only batch mode components, then you can disable SCBroker to prevent it from listening on a TCP port.

SCBroker listens on a configurable, static port for new connection requests from the Siebel Application Interface. The parameter Static Port Number (alias PortNumber) defines the port that SCBroker monitors; the default value is 2321. You specify this port number for SCBroker in the Siebel Management Console, when you configure the Siebel Server and when you configure the Siebel Application Interface. After a request is received, SCBroker distributes it to the appropriate instance of an Application Object Manager running on the Siebel Server. The SCBroker component uses a connection forwarding algorithm to forward the socket to the Application Object Manager processes, according to the setting of the component parameter Connection Forward Algorithm for SCBroker (alias ConnForwardAlgorithm).

ConnForwardAlgorithm, which is a hidden parameter, supports the following possible settings:

- **LL.** The least-loaded algorithm (default behavior) balances incoming Application Object Manager login requests. It identifies which Application Object Manager process is handling the least number of tasks and assigns that process to handle the session. If SCBroker determines that an Application Object Manager process is not responding to a request, then it sends subsequent requests to the next available Application Object Manager process (using least-loaded algortihm).
- **RR.** The round-robin algorithm distributes all of the Application Object Manager login requests to the next Application Object Manager process in a round-robin fashion, that is, equal loads distributed in order and without priority. If SCBroker determines that an Application Object Manager process is not responding to a request, then it sends subsequent requests to the next available Application Object Manager process (using round-robin algortihm).

For more information about SCBroker and about load balancing, see *Siebel Deployment Planning Guide* and *Siebel Installation Guide* .

About Other System Management Components

This topic describes the other server components that make up the System Management (alias System) and the Auxiliary System Management (SystemAux) component groups. It includes the following information:

- *About Server Tables Cleanup (SvrTblCleanup)*

- *About Siebel Administrator Notification (AdminNotify)*
- *About Siebel Server Scheduler (SrvrSched)*

About Server Tables Cleanup (SvrTblCleanup)

Server Tables Cleanup (alias SvrTblCleanup) is a component that deletes the completed and expired Server Request records. The parameter Sleep Time (alias SleepTime) controls how often the cleanup occurs. The default value for Sleep Time is 300 seconds (5 minutes).

By default, the Server Tables Cleanup component is enabled on all of the Siebel Servers in your Siebel Enterprise Server. However, you only have to run one instance of this component, because it deletes the completed and expired server request records for all of the Siebel Servers in the Siebel Enterprise Server from the Siebel database. For this reason, you can disable other instances of this component on other Siebel Servers in the Siebel Enterprise Server.

The Server Tables Cleanup component is part of the Auxiliary System Management component group.

Server Tables Cleanup removes deletes completed and expired requests from the following tables:

- S_SRM_REQUEST
- S_SRM_REQ_PARAM
- S_SRM_DATA
- S_SRM_TASK_HIST
- S_OM_TEMP
- S_CM_MSG_STAT

About Siebel Administrator Notification (AdminNotify)

Siebel Administrator Notification (alias AdminNotify) is a batch mode component that notifies the Siebel administrator when problems are detected on the Siebel Server or its running components. For more information about component notification, see *About System Alert Notification*.

The Siebel Administrator Notification component is part of the Auxiliary System Management component group.

About Siebel Server Scheduler (SrvrSched)

Siebel Server Scheduler (alias SrvrSched) is a background mode component supports the running of the Siebel Server and server components by spawning component processes as requested by the Siebel Server. No entries for the Siebel Server Scheduler component appear in the Siebel Server log file. Instead, entries appear for the component for which Siebel Server Scheduler spawns a process. A network message eventually assigns the process to the component it is supposed to run. The process loads the component and runs it. The Siebel Server Scheduler component is part of the System Management component group.

> **CAUTION:** Do not modify the Siebel Server Scheduler component without instructions from Global Customer Support. For help modifying Siebel Server Scheduler, create a service request (SR) on My Oracle Support. Alternatively, you can phone 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.

Note: Because of the nature of the Siebel Server Scheduler component, the Siebel Server Scheduler task IDs that appear in the log files do not have an appropriate entry in the Administration - Server Management screen. For the same reason, no entry appears in the Components view of the Administration - Server Configuration screen for this component.

Backing Up and Restoring the Siebel Gateway Registry

This topic describes how to back up the Siebel Gateway registry and restore the Siebel Gateway registry from a backup. For more information about the Siebel Gateway registry, see *About the Siebel Gateway*. See also *Siebel Installation Guide* .

This topic contains the following information:

- *Backing Up the Siebel Gateway Registry*
- *Restoring the Siebel Gateway Registry*

Backing Up the Siebel Gateway Registry

To back up the Siebel Gateway registry, perform the steps in the procedure that follows. It is strongly recommended to back up the Siebel Gateway registry periodically for safety reasons, so that, if something goes wrong, you will be able to restore the Siebel Gateway registry from a backup, as described in *Restoring the Siebel Gateway Registry*.

It is recommended to back up the Siebel Gateway registry at regular intervals, and before deploying Siebel Gateway clustering into your topology. Retain the backups, which you can use to restore the Siebel Gateway registry data in scenarios identified as caused by corruption in the Siebel Gateway registry. One of the known scenarios where a restore is required is if Siebel Gateway cluster deployment fails and the administrator experiences inconsistency in accessing profiles and deployments in Siebel Management Console. Where you have already deployed Siebel Gateway clustering, then you create the backup for *only one* of the deployed instances. For more information about Siebel Gateway clustering, see *Siebel Installation Guide* .

This topic is part of *Backing Up and Restoring the Siebel Gateway Registry*.

To back up the Siebel Gateway registry

1. (Optional) Shut down the Siebel CRM deployment, as described in *Starting and Shutting Down a Siebel CRM Deployment*.
2. For the instance of Siebel Gateway for which you are backing up the registry, do the following:
   a. Navigate to the directory `GTWYSRVR_ROOT\registry\conf`. (Prior to Siebel CRM 21.4 Update, the `registry` directory was named `zookeeper`.)
   b. Note the dataDir path in the registry.cfg file. (Prior to Siebel CRM 21.4 Update, this file was named zoo1.cfg.)
   c. Copy the `version-2` directory to another name, such as one that includes the current date.
3. (If necessary) Restart the Siebel CRM deployment, as described in *Starting and Shutting Down a Siebel CRM Deployment*.

Restoring the Siebel Gateway Registry

To restore the Siebel Gateway registry from the backup, perform the steps in the procedure that follows.

> **CAUTION:** Do not perform the task of restoring the Siebel Gateway registry unless it is strictly necessary to resolve a problem. You must have first created a valid backup, as described in *Backing Up the Siebel Gateway Registry*.

Where you have already deployed Siebel Gateway clustering, then you restore the same backup for all of the deployed instances. For more information about Siebel Gateway clustering, see *Siebel Installation Guide* .

This topic is part of *Backing Up and Restoring the Siebel Gateway Registry*.

To restore the Siebel Gateway registry

1. Shut down the Siebel CRM deployment, including all Siebel Gateway cluster nodes, as described in *Starting and Shutting Down a Siebel CRM Deployment*.
2. For each deployed instance of Siebel Gateway, including all Siebel Gateway cluster nodes, do the following:
   a. Navigate to the directory `GTWYSRVR_ROOT\registry\conf`. (Prior to Siebel CRM 21.4 Update, the `registry` directory was named `zookeeper`.)
   b. Note the dataDir path in the registry.cfg file. (Prior to Siebel CRM 21.4 Update, this file was named zoo1.cfg.)
   c. Remove the `version-2` directory (if it exists).
   d. Into this same location, copy the backup directory that you created in *Backing Up the Siebel Gateway Registry*.
   e. Rename this directory to `version-2`.
3. Restart the Siebel CRM deployment, as described in *Starting and Shutting Down a Siebel CRM Deployment*.

Administering the Siebel File System

This topic provides background information and administration tasks applicable to the Siebel File System. This topic contains the following information:

- *About the Siebel File System*
- *About the File System Upload and Download Process*
- *Partitioning the Siebel File System*
- *Cleaning Up the Siebel File System*

About the Siebel File System

The Siebel File System is a shared directory or a set of directories that contain the physical files used by the Siebel clients. All of the File System directories must be network-accessible to the Siebel Server. You can create each File System directory on a server computer where you have installed a Siebel Server, or on another network server that can share the directory, so that the File System directories are available to the Siebel Server. For more information about the requirements for networked file systems, see the third-party documentation.

To gain access to files, Web clients connect to the appropriate Siebel Server to request file uploads or downloads. The Siebel Server then accesses the Siebel File System using the File System Manager (alias FSMSrvr) component. File System Manager processes these requests through interaction with the Siebel File System directories. For more information about data transfer, see *About the File System Upload and Download Process*.

At the server component level, most server components, including all Application Object Managers, access the Siebel File System through the File System Manager server component when administering attachments. Application Object Managers, however, access the Siebel File System directly when saving user preference files.

When using Siebel Developer Web Client for administrative tasks, you might want to connect directly to the Siebel File System without going through the File System Manager.

Files stored in the Siebel File System are compressed at the Siebel Server-level and appended with the extension .saf. (The file size displayed in the GUI represents the size of the compressed .saf file, not the actual file size.) The Siebel File System storage locations of the compressed files are set by the enterprise parameter Siebel File System (alias FileSystem). For more information about this parameter, see *Siebel Enterprise Server Parameters*. The files stored in the Siebel File System are not directly accessible by users and must be retrieved (and decompressed) by the user through normal Siebel Web Client operations only.

Files stored in the Siebel File System are always compressed. That is, you cannot disable the compression feature of the Siebel File System.

You can exclude certain types of files from being saved into the Siebel File System, based on their file extensions. For more information about setting system preferences to enable this feature and to specify the file extensions to be excluded, see *Siebel Security Guide* .

Note: Virus checking is not supported within the Siebel File System.

This topic is part of *Administering the Siebel File System*.

Related Topics

About the File System Upload and Download Process

Partitioning the Siebel File System

Cleaning Up the Siebel File System

Related Books

For information about creating the Siebel File System, see *Siebel Installation Guide* .

For information about deployment options for the Siebel File System, see *Siebel Deployment Planning Guide* .

For information about securing the Siebel File System, see *Siebel Security Guide* .

About the File System Upload and Download Process

This topic describes what happens when files are uploaded to or downloaded from the Siebel File System.

This topic is part of *Administering the Siebel File System*.

About the File System Upload Transfer Process

When a user saves a file or attachment to be written to the Siebel File System, the file is copied from the user's hard drive and transferred to the Siebel Server. The data transfer protocol for file transfer matches that of the Web client browser to Siebel Application Interface, for example, HTTP or HTTPS. The File System Manager (alias FSMSrvr) component compresses the file, and then stores the compressed file in the Siebel File System. The compression and naming convention of the files is automated by FSMSrvr.

About the File System Download Transfer Process

When a Siebel application user accesses a file (for example, a PDF document) that is stored in the Siebel File System, a file or attachment download request is received by the FSMSrvr component of the Siebel Server. This component interacts with the Siebel File System directories to retrieve and send the compressed file back to the user's Web browser. As with the file upload process, the data transfer protocol for file transfer matches that of the Web client browser to Siebel Application Interface. The compressed file is decompressed by the user's Web browser, where the file can be reviewed or saved.

In some cases, the file is decompressed by the FSMSrvr component and sent to the user's Web browser in an uncompressed format. An uncompressed file is sent back to the Web browser in the following cases:

- The parameter Compressed File Download (alias CompressedFileDownload) is set to False. You configure this parameter in the Siebel Server Component Parameters view. For information about this task, see *Configuring Siebel Server Component Parameters*.
- The CompressedFileDownload parameter is set to False in the application configuration file for a Siebel Mobile Web Client. (If this parameter is not already in the configuration file, then you can add it to the [InfraUIFramework] section of the file.)
- The Web browser does not support compressed files, which is determined by looking at the request header.
- The file has the extension .zip, .z, .tgz, .gz, .gif, .jpg, or .jpeg.

Partitioning the Siebel File System

This topic describes how to perform the optional task of partitioning the Siebel File System.

This topic is part of *Administering the Siebel File System*.

This topic contains the following information:

- *About Partitioning the Siebel File System*
- *Partitioning the File System Directories Using the sfspartition Utility*
- *Parameters for the sfspartition Utility*

About Partitioning the Siebel File System

This topic describes how to perform the optional task of partitioning the Siebel File System.

This topic is part of *Partitioning the Siebel File System*.

Partitioning the Siebel File System allows you to store larger volumes of data on multiple devices. The original Siebel File System might use a single directory or might already use multiple directories on multiple devices or partitions.

In general, the term *partitioning*, as used in this topic, refers to running the `sfspartition` utility, which is provided for the purpose of adding one or more network directories to an existing Siebel File System and distributing the existing files

among all of the participating directories. You can add each new directory on the same device as an existing directory or add it on a different device or partition in order to expand the overall capacity of the Siebel File System. (You must consider the future growth of the volume of data when you plan how to organize the file system directories.)

You can also use `sfspartition` to remove one or more existing directories from service for the Siebel File System, provided that the overall file system capacity remains sufficient. To partition your Siebel File System, you first update the value of the enterprise parameter Siebel File System (alias FileSystem) so it specifies all of the network directories that you want to use for the Siebel File System, delimited by commas. You then run the partitioning utility and specify both the original directories containing the existing files and the updated target directories, corresponding to the updated FileSystem parameter value. The sfspartition utility distributes the files in the Siebel File System evenly across the target directories. The utility logs information into a file named sfspartition.log, which is located in the log directory within the Siebel Server root directory.

When the File System Manager component (alias FSMSrvr) starts, it verifies the existence of all of the file system directories specified using the FileSystem parameter. When new file attachments are inserted, FSMSrvr distributes them across these directories. If a file system directory is unavailable, then FSMSrvr logs an error message in the FSMSrvr log file and tries to write the file attachment to the next available directory. If no file system directory is available, then FSMSrvr terminates and writes an error message to the FSMSrvr log file.

In order to maintain the even distribution of files across file system directories, you must run the partitioning utility every time that you update the value of the FileSystem parameter, for example, if you add or remove a file system directory. The procedure in this topic describes in detail how to perform this task.

Before you partition your Siebel File System, note the following additional deployment options:

- Mobile Web Client. A Mobile Web Client's configuration file must refer to a single directory location, unless you configure it to use the server-based data source.
- Replication Manager requirements. Partitioning is supported on replicated nodes. For more information about replication, see *Siebel Remote and Replication Manager Administration Guide* .

Partitioning the File System Directories Using the sfspartition Utility

To partition the Siebel File System directories, run the partitioning utility `sfspartition`, as described in the following instructions.

This topic is part of *Partitioning the Siebel File System*.

The partitioning utility is named `sfspartition.exe` on Microsoft Windows or `sfspartition` on UNIX operating systems. This utility is located in the `bin` directory within the Siebel Server root directory.

> **Note:** Where necessary, before you run the `sfspartition` utility, you must manually create any file system directories (such as the examples `siebelFS1`, `siebelFS2`, and `siebelFS3`, and so on) and subdirectories (such as `att`, `attmp`, and so on) on each target file system and grant the appropriate permissions to all of these directories. All of the file system directories must be accessible to all of the applicable Siebel Servers using the notation by which they are represented in the value of the FileSystem parameter.

For more information about creating the Siebel File System, see *Siebel Installation Guide* .

For more information about the parameters you can use for the `sfspartition` utility, see *Parameters for the sfspartition Utility*.

To partition the file system directories using sfspartition

1. Where necessary, create any new directories or partitions that you will use with the Siebel File System, create required subdirectories, and grant the appropriate permissions to all of the directories. Then verify access to these directories.
2. Note the current value of the enterprise parameter Siebel File System (alias FileSystem) for later reference.

   > **Note:** You must note this value because you will use this information later when you specify the source directories by using the /O parameter when you run the `sfspartition` utility.

3. Set the value of the FileSystem parameter to include all of the directories that you want to use for the Siebel File System. Separate each directory with a comma (with no spaces), as in the examples that follow. Note the updated parameter value for later reference.

   > **Note:** You must note this value because you will use this information later when you specify the target directories by using the /F parameter when you run the `sfspartition` utility.

   For example, on UNIX, you might specify this value:

   ```
   /export/home/siebelFS1,/export/home/siebelFS2,/export/home/siebelFS3
   ```

   For example, on Microsoft Windows, you might specify this value:

   ```
   \\\\server1\\siebelFS1,\\\\server1\\siebelFS2,\\\\server2\\siebelFS3
   ```

   > **Note:** In this example, note that each backslash is doubled compared to the usual notation for such shared directories. For example, `\\server1\siebelFS1` must be represented as `\\\\server1\\siebelFS1`. Alternatively, each file system directory can be represented using a mapped drive letter by which the directory can be accessed from each Siebel Server, such as `D:\\siebelFS1` (note that each backslash must be doubled in this scenario also).

   > **CAUTION:** You must specify the file system directories in the same order for the FileSystem parameter and for the `sfspartition` utility. If you specify the directories using a different order, then the Siebel File System files might not be accessible after you use `sfspartition`.

4. If it is not already set, then set the ServerDataSrc named subsystem parameter DSFileSystem to *FSM*.

   For information about configuring named subsystem parameters, see *Configuring Siebel Enterprise Server Named Subsystem Parameters*.
5. Restart the Siebel Server after updating the FileSystem and DSFileSystem parameter values.
6. Open a command prompt and change the directory to the `bin` subdirectory within the Siebel Server root directory.
7. Run `sfspartition` using parameters listed in this topic, as in the examples that follow.

   The following example for UNIX distributes the files from one file system directory into three directories, corresponding to the updated value of the FileSystem parameter from this procedure:

   ```
   sfspartition /O /export/home/siebelFS /F /export/home/siebelFS1,/export/home/siebelFS2,/export/home/
   siebelFS3 /H Y
   ```

   > **Tip:** Depending on how these network directories were created or mounted for use in UNIX environments, they might be on the same server or on different servers.

The following example for Microsoft Windows distributes the files from one file system directory on server1 into three directories on server1 and server2, corresponding to the updated value of the FileSystem parameter from this procedure:

```
sfspartition /O \\server1\siebelFS /F \\server1\siebelFS1,\\server1\siebelFS2,\\server2\siebelFS3 /H Y
```

Parameters for the sfspartition Utility

The following table describes the parameters for the **sfspartition** utility.

This topic is part of *Partitioning the Siebel File System*.

Parameter	Value	Description	Required?
/O	Paths for existing source directories	Set this value to the paths of the existing source directories for the file system. Separate multiple directories using commas, with no spaces. If any of the paths themselves contain a space, then enclose the parameter value in double quotes. Whether you must append **att** to each source directory depends on how you use the /H parameter: • If you use /H Y, then do not append **att** to each source directory that you specify by using /O. (The purpose of /H Y is to eliminate the need to append **att**.) • If you use /H N (or omit /H), then you must append **att** to each source directory that you specify by using /O. The utility looks both in the specified source directories and in the **att** subdirectories to find the files to be distributed to the specified target directories.	Yes
/F	Paths for target directories	Set this value to the paths of the target directories for the file system. Separate multiple directories using commas, with no spaces. If any of the paths themselves contain a space, then enclose the parameter value in double quotes. (Use the same value as the value of the FileSystem parameter.) Whether you must append **att** to each target directory depends on how you use the /H parameter: • If you use /H Y, then do not append **att** to each target directory that you specify by using /F. (The purpose of /H Y is to eliminate the need to append **att**.) • If you use /H N (or omit /H), then you must append **att** to each target directory that you specify by using /F. Otherwise, files will not be distributed to the **att** subdirectories of the target directories and will be inaccessible to clients.	Yes
/H	Y or N	Set /H Y if you want the utility to automatically append **att** to each source and target directory that you specify by using /O and /F. (Do not append **att** when you specify these directories.) Set /H N (or omit /H) if you do not want the utility to automatically append **att** to each source and target directory	No

Parameter	Value	Description	Required?
		that you specify by using /O and /F. (Append `att` when you specify these directories.)	

Cleaning Up the Siebel File System

This topic describes how to clean up the Siebel File System by removing orphan records using the Siebel File System cleanup utility, `sfscleanup`. Orphan records are those that remain if a user deletes a parent record in the application that has associated child records. The child records are not deleted from the Siebel File System with the parent record and so you must remove them by using file system cleanup utility.

The Siebel File System cleanup utility is named `sfscleanup.exe` on Microsoft Windows or `sfscleanup` on UNIX operating systems. This utility is located in the `bin` directory within the Siebel Server root directory.

The `sfscleanup` utility processes records for every file in the file attachment directories (the `att` subdirectories) of the specified Siebel File System directories and performs one of several operations to each record and file, depending on the file type and on the parameters that you set. Optionally, you can run `sfscleanup` for a limited period of time and resume the operation again later.

For information about running `sfscleanup`, including descriptions of the file types and the associated operations performed by `sfscleanup` during processing, and the run-time parameters you can use for the utility, see the information that follows.

This topic is part of *Administering the Siebel File System*.

This topic contains the following information:

- *Cleaning Up the File System Directories Using the sfscleanup Utility*
- *Parameters for the sfscleanup Utility*
- *Unzipping Siebel File System Files from the Command Line*

Cleaning Up the File System Directories Using the sfscleanup Utility

This topic describes how to clean up the Siebel File System directories by removing orphan records using the Siebel File System cleanup utility, `sfscleanup`.

This topic is part of *Cleaning Up the Siebel File System*.

To clean up the file attachment directory using sfscleanup

1. At the command prompt, change directory to the `bin` subdirectory within the Siebel Server root directory.
2. Run `sfscleanup` using parameters listed here, in a command like the following example:

   ```
   sfscleanup /U sadmin /P pwd /F \\server1\files /X \\server1\logs\sfscleanup.log
   ```

   For more information about the parameters you can use for the `sfscleanup` utility, see *Parameters for the sfscleanup Utility*.

About the sfscleanup Log File

If you specified an output file using the /X parameter, then **sfscleanup** generates a log file listing the operations that were performed. The output file is a tab-delimited text file that contains the following columns:

- **File Name.** This column lists the name of each file that was processed.
- **File Type.** This column lists the type of each file that was processed.
- **Operation.** This column lists the type of operation that was performed during processing. A table later in this topic provides descriptions of each operation.

File Types and Operations for the sfscleanup Utility

The following table lists the possible file types and the associated operation performed by **sfscleanup** during processing. The Operation column lists the type of operation that was performed during processing.

File Type	Description	Operation
CURRENT	The file has a corresponding record in the file attachment database table.	KEPT
NEW	The file is less than one hour old. The **sfscleanup** utility does not check for the file in the file attachment database table.	KEPT
ORPHAN	The file does not have a corresponding record in the file attachment database table. If you used the /M parameter to set a move directory, then the operation performed is MOVED, not DELETED.	DELETED
INVALID	The file (or directory) is not a file attachment. If **sfscleanup** tries to delete a subdirectory that is not empty, then the operation errors out. Review the files contained within the directory before deleting them. If you set the /G parameter to Y, then the operation performed is DELETED, not KEPT.	KEPT
ANCIENT	The file has an associated record in the database with a different revision number. If you set the /N parameter to Y, then the operation performed is either MOVED (if you used the /M parameter to set a move directory) or DELETED, not KEPT.	KEPT

The operations mentioned in the previous table are described in the following table.

Operation	Description
KEPT	The file was kept.
DELETED	The file was deleted.

Operation	Description
MOVED	The file was moved to the directory specified by the /M parameter. Files are moved if you used the /M parameter.
KEPT_DIR	The item was kept because it was a directory and requires manual processing.
KEPT_ERROR	The file was kept because an error occurred while trying to move or delete the file.

Parameters for the sfscleanup Utility

The following table describes the parameters for the **sfscleanup** utility. More information about some of the parameters is provided after the table.

This topic is part of *Cleaning Up the Siebel File System*.

Parameter	Value	Description	Required?
/U	Username	User name ID.	Yes
/P	Password	User name password.	Yes
/C	ODBC_data_source	Set this value to the ODBC data source. The default value is the setting of the environment variable SIEBEL_DATA_SOURCE.	No
/D	Siebel_table_owner	Set this value to the Siebel table owner. The default value is the setting of the environment variable SIEBEL_TABLE_OWNER.	No
/F	Paths for file system directories	Set this value to the paths for the file system directories. Separate multiple directories using commas, with no spaces. If any of the paths themselves contain a space, then enclose the parameter value in double quotes. (Use the same value as the value of the FileSystem parameter.) Whether you must append **att** to each directory depends on how you use the /H parameter: • If you use /H Y, then do not append **att** to each directory that you specify using /F. (The purpose of /H Y is to eliminate the need to append **att**.) • If you use /H N (or omit /H), then you must append **att** to each directory that you specify using /F. The utility looks both in the specified directories and in the **att** subdirectories to find the files to be cleaned up.	Yes
/X	Path for output file	Set this value to the path for the output file.	No

Parameter	Value	Description	Required?
/M	Path for move directory	Set this value to the path for the directory where discarded files are to be moved.	No
/N	Y or N	Determines whether old versions of file attachments are to be removed. To remove old versions, set this value to Y. The default value is N.	No
/R	Y or N	Set this value to Y to generate only a report file. If it is set to Y, then the report file contains only the columns File Name and File Type. The default value is N.	No
/H	Y or N	Set /H Y if you want the utility to automatically append **`att`** to each directory that you specify using /F. (Do not append **`att`** when you specify these directories.) Set /H N (or omit /H) if you do not want the utility to automatically append **`att`** to each directory that you specify using /F. (Append **`att`** when you specify these directories.)	N
/G	Y or N	Set this value to remove garbage files or non-Siebel files. The default value is N.	No
/Q	Y or N	Set /Q Y if you want the utility to perform a query by file attachment records. This parameter allows you to run the utility for a limited period of time and provides other ways to manage how the utility runs. The default value is N. **Note:** When you are using /Q Y, you can also use the parameters /I, /O, /S, and /T. Otherwise, these parameters have no effect.	N
/I	Number of file IDs	Set /I to the number of file attachment records to query. The default value is 300. The utility processes records and files in batches based on the specified number of records. After those files have been processed, the utility processes more records and files in another batch. **Note:** This parameter has an effect only if you are using /Q Y.	N
/O	Y or N	Use /O Y when you want the utility to use an OR clause to constrain the query row IDs, like this: **`(ROW_ID = 'Id1' OR ROW_ID = 'Id2' OR ...)`**. The default value is Y. Use /O N to instead use a clause like this: **`ROW_ID IN ('Id1','Id2',...)`**.	N

Parameter	Value	Description	Required?
		Note: This parameter has an effect only if you are using /Q Y. It determines the internal query executed in Siebel database. No row ID is entered as an argument.	
/S	Y or N	Use /S Y to resume the previous run, from the next unprocessed record, where information about the last processed record is available in a temporary directory. The default value is N. Use /S N (or omit /S) to instead start a new run. **Note:** This parameter has an effect only if you are using /Q Y.	N
/T	Number of minutes	Set /T to the number of minutes to run the query. When the utility reaches that time, the last processed file attachment record is noted in a temporary directory and the utility exits. Later, you can resume the previous run by using /S Y. By default, the utility runs to completion, until all of the records and files are processed. **Note:** This parameter has an effect only if you are using /Q Y.	N

More Information About Some sfscleanup Parameters

The following provides more information about some of the parameters for the `sfscleanup` utility.

- **/N.** By default, old file revisions are kept. Such files are marked ANCIENT in the log, and represent old revisions of an existing attachment record. That is, their row ID matches with the database record but not the file revision number. To delete such files, set the /N parameter to Y.
- **/G**. If the file system contains files that were not created by the File System Manager component (alias FSMSrvr), then their deletion or move is controlled by the /G parameter. This parameter includes non-Siebel files or directories. By default these files are not deleted. The directories are not affected or moved by `sfscleanup`.
- **/Q.** By default, the `sfscleanup` utility processes all of the files in the file attachment directories in a single long-running operation. For a Siebel File System that includes a very large number of files, such an operation might affect performance or inconvenience production users.

  Alternatively, the /Q parameter allows you, for example, to process files in batches based on a query of a given number or file attachment records (by using /I), to modify how the utility queries these records (by using /O), to run the utility for a specific period of time (by using /T), and to resume a run later where it left off (by using /S). For example, you might decide to run the `sfscleanup` utility only when most of your users are not logged in.

Unzipping Siebel File System Files from the Command Line

If a file is orphaned and no longer has a link to a record in a Siebel Attachment table in the database, it will no longer be accessible through the UI. To retrieve the file, it is possible to use the `sseunzip80` utility found in the Siebel Server's **bin** folder.

```
sseunzip80 FILENAME.SAF DestinationFile
```

For example, to unzip the file `s_doc_corr_0-5704_0-yz-5.saf` to **MyDocument.docx**, enter `sseunzip80 ...\fs\att\s_doc_corr_0-5704_0-yz-5.saf MyDocument.docx`

> **Note:** The `sseunzip80` utility does not know what the original name and file extension are, so an administrator using this utility would have to guess at the correct file type, such as DOCX, PDF, XLSX, etc.

Rolling Back the Siebel Runtime Repository to a Prior Version

You can use the Siebel runtime repository version rollback feature in the Siebel application to revert to the last known good version of a deployed application's metadata (Siebel repository definitions). For more information about this feature, see *Siebel Applications Administration Guide* .

> **Note:** Siebel runtime repository version rollback applies only to runtime repository (RR) environments, such as QA or production. This feature is available in Siebel CRM 20.3 Update and later releases.

9 Application Object Manager Administration

Application Object Manager Administration

This chapter explains how to configure, deploy, and administer Application Object Managers to support Siebel Web Clients. This chapter also explains Application Object Manager concepts to provide useful background information. It includes the following topics:

- *About the Siebel Application Object Manager*
- *Configuring the Application Object Manager Environment*
- *About Application Object Manager Parameters*
- *Administering the Application Object Manager*

About the Siebel Application Object Manager

Siebel Application Object Manager (Application Object Manager) components host the Business Objects layer and Data Objects layer of the Siebel architecture. The Siebel Web Clients host the Siebel application user interface layer. The Application Object Manager supports Siebel Web Client connections and handles multiple users simultaneously by making requests to the Siebel Server on behalf of the clients.

Application Object Managers are hosted as components in the installed Siebel Server and run on the Siebel Server computer (sometimes called the application server computer). The Siebel Server provides the infrastructure for an Application Object Manager to serve multiple Siebel Web Client users. Multiple Application Object Manager components can run on a single Siebel Server installation. Application Object Manager components can be configured to run as multithreaded processes in the Siebel Server. Like other Siebel Server components, you can administer Application Object Manager components using the Siebel Server Manager.

Application Object Managers communicate with clients using the TCP/IP protocol through the Siebel Application Interface. Communication between the Siebel Application Interface and the Application Object Manager can be compressed and encrypted. An independent session is established to serve incoming connect requests from each client. Subsequent requests from clients are directed to the same Application Object Manager tasks until the sessions are terminated.

After startup, Application Object Managers do not achieve their full run-time environments until after the first connect, therefore, leading to possible delays during the first connection. For more information about how the Siebel Web Clients and Application Object Managers communicate, see *Siebel Performance Tuning Guide* .

The Siebel runtime repository is installed as part of each Siebel database installation. User preferences set and saved by Siebel Web Client users are saved on the Siebel Server.

Note: If you are running the Siebel Server in a UNIX environment, then Application Object Managers support Siebel eScript, but not Siebel Visual Basic.

About Application Object Manager Memory Allocation

The Application Object Manager caches information in two different ways. Some information is cached and used by every connection and other information is stored for each user connection.

Memory allocation for the Application Object Manager can be broken into three areas:

- **User memory.** The user area maintains specific information about each user's session. Typically, each user uses 3 MB to 4 MB of memory, although the memory required depends on the Siebel application in use. This memory is released when the task is completed.
- **Shared memory between users.** Shared memory is for common structures used by every user and is the largest segment of Application Object Manager memory. It contains definitions for business objects, business components, controls, and other metadata items from the Siebel repository. This memory is loaded as needed and remains loaded for the life of the process. Application Object Manager processes commonly use more than 150 MB.
- **Administrative memory.** The administrative area of memory used by Application Object Manager manages the component itself. This memory is relatively small and is used to manage communication between the Application Object Manager and other Siebel Server components. It runs the listener and coordinates threads and tasks.

Configuring the Application Object Manager Environment

This topic describes how to configure the environment for your Application Object Manager components.

To configure the Application Object Manager environment

1. Make sure that Application Object Managers are included as part of the Siebel Server installations you plan to use.

   Configuring a Siebel Server defines the Application Object Manager components for this Siebel Server. For information about installing and initially configuring the Siebel Server, see *Siebel Installation Guide* .
2. Verify that the customized Siebel runtime repository is included in the Siebel database that serves the modified application.

   It is strongly recommended that no other user besides the application developer customize the Siebel runtime repository.
3. Configure the Application Object Manager components by setting the parameters that control:
   - Application name to run (configuration file)
   - Language code
   - Compression setting
   - Encryption setting
   - Number of processes for each component
   - Number of threads or tasks for each process

- Session Manager parameters

For more information about this subject, see *About Application Object Manager Parameters*.

4. Restart the Siebel Server system service to automatically register the Application Object Manager services that you configured in the previous step.

About Application Object Manager Parameters

At startup, Application Object Manager components accept several parameters that determine their behavior.

This section includes the following information:

- *Application Object Manager Parameters in Server Manager*
- *Application Object Manager Named Subsystem Parameters*

Parameters that affect the operation of Application Object Managers can be modified as follows:

- Component parameters for the Application Object Manager using the Siebel Server Manager. For more information about this task, see *Configuring Siebel Server Component Parameters*.
- Named subsystem parameters using Siebel Server Manager. For more information about this task, see *Configuring Siebel Enterprise Server Named Subsystem Parameters*.

Application Object Manager Parameters in Server Manager

This topic provides a partial list of component-specific and generic parameters that you set for the Application Object Manager. You configure Application Object Manager parameters in the same manner as those for any server component.

To configure Application Object Manager parameters using the Server Manager GUI, see *Configuring Siebel Server Component Parameters*. To configure Application Object Manager parameters using the Server Manager command-line interface, see *Parameter Management Commands*.

See the following parameters:

- Compression Type (alias Compress)
- Encryption Type (alias Crypt)
- Error Flags (alias ErrorFlags)
- Language Code (alias Lang)
- Log Print Timestamp (alias LogTimestamp)
- Maximum MT Servers (alias MaxMTServers)
- Maximum Tasks (alias MaxTasks)
- Minimum MT Servers (alias MinMTServers)
- Multi-Threaded (alias Threaded)
- Number of lines after which to flush the log file (alias LogFlushFreq)
- Password (alias Password)
- Trace Flags (alias TraceFlags)

- User Name (alias Username)
- Use Shared Log File (alias LogUseSharedFile)

For a description of each parameter, see *Siebel Enterprise, Server, and Component Parameters*.

For information about locale-specific Application Object Manager parameters, including information about regional standards for currency, time, date, and so on, see *Siebel Global Deployment Guide* .

Application Object Manager Named Subsystem Parameters

The Application Object Manager can maintain several different values for a particular parameter by using named subsystems.

For more information about named subsystems and named subsystem parameters, see *About Named Subsystem Parameters*. For information about configuring Application Object Manager named subsystem parameters, see *Configuring Siebel Enterprise Server Named Subsystem Parameters*.

To configure named subsystems using the Siebel Server Manager command-line interface, see *Named Subsystem Management Commands*.

Named Subsystems Used by Application Object Managers

The following table provides a partial list of named subsystems used by Application Object Managers.

Named Subsystem	Alias	Type	Description
DataMart Datasource	DataMart	InfraDatasources	Datamart data source used by Application Object Manager components
Gateway Datasource	GatewayDataSrc	InfraDatasources	Gateway data source used by Application Object Manager components
Object Manager Cache	ObjMgrCache	InfraObjMgrCache	Cache parameters for Application Object Manager components
Server Datasource	ServerDataSrc	InfraDatasources	Server data source used by Application Object Manager components
LDAP Security Adapter	LDAPSecAdpt	InfraSecAdpt_LDAP	LDAP security adapter used for Application Object Manager authentication with directory servers

Named Subsystem Parameters

The following table provides a partial list of named subsystem parameters.

Parameter Alias	Named Subsystem	Data Type	Description
DSConnectString	GatewayDataSrc	String	Specifies the host name of the Siebel Gateway. The value $(GatewayAddress) for the parameter ConnectString of the GatewayDataSrc section of the Application Object Manager's configuration file is replaced at run time with the value for the named subsystem parameter DSConnectString. An incorrect setting for DSConnectString results in server administration being inaccessible from the Siebel Web Client.
DSConnectString	ServerDataSrc	String	Specifies the database connection information. On Oracle Database (native), this information is the TNS alias from tnsnames.ora. On Microsoft SQL Server and IBM DB2, this information is the ODBC Datasource Name.
DSDockedFlg	ServerDataSrc	Boolean	When you create a new named subsystem for an Application Object Manager, you must set this parameter to True.
DSFileSystem	ServerDataSrc	String	Set this parameter to *FSM* to allow the use of the File System Manager server component for standard Siebel Web Clients.
DSMaxCursorSize	ServerDataSrc, GatewayDataSr, or DataMart	Integer	Sets the total number of rows that can be returned in a result set. The parameter MaxCursorSize is set for the Application Object Manager component by using the named subsystem parameter DSMaxCursorSize. (The applicable subsystem corresponds to the component parameter OM - Data Source.) This parameter is valid only with IBM DB2 for OS/390 and z/OS. For more information, see the description of the MaxCursorSize parameter in *Parameters for Individual Data Source Sections*.
DSPreFetchSize	ServerDataSrc, GatewayDataSr, or DataMart	Integer	Sets the number of rows that the Siebel application reads initially as part of a query execution. The parameter PreFetchSize is set for the Application Object Manager component by using the named subsystem parameter DSPreFetchSize. (The applicable subsystem corresponds to the component parameter OM - Data Source.) This parameter is valid only with IBM DB2 for OS/390 and z/OS.

Parameter Alias	Named Subsystem	Data Type	Description
			For more information, see the description of the PreFetchSize parameter in *Parameters for Individual Data Source Sections*.
DSEnterpriseServer	ServerDataSrc, GatewayDataSr	String	Specifies the name of the Siebel Enterprise Server used by various named subsystems. The value in the Application Object Manager is replaced at run time with the value set for the named subsystem.
DSRequestServer	ServerDataSrc	String	Specifies where asynchronous calls are redirected, for example when using Interactive Assignment. Set this value to the logical name of the Siebel Server, not the computer name. The default value is null.
DSTableOwner	All named subsystems of type InfraDatasources	String	Specifies the table owner for this data source. Application Object Manager server components, such as Call Center Object Manager, read the value of this parameter. Siebel Server infrastructure and system management components, such as Server Request Broker (SRBroker), read the value of the TableOwner enterprise parameter.
DSDisableExecuteRetry	ServerDataSrc	Boolean	When set to True, this parameter prevents the Application Object Manager from resending a query to the database layer if an error occurs such as a network error or session kill.

Administering the Application Object Manager

You can monitor Application Object Managers at the following levels:

- The server level using Siebel Server
- The component level using Application Object Manager components
- The task level using Application Object Manager tasks

At each of these levels, you can do the following:

- Use the server administration views to monitor the following:
  - State values
  - Statistics
  - Log files

- Use the Siebel Server Component Parameters view to set the component-specific parameters for the Application Object Manager.
- Start, stop, pause, or resume any Application Object Manager tasks.

At the component event level, you can enable SQL tracing to view the SQL that is generated for the specified Application Object Manager. You can enable SQL spooling on the Application Object Manager task by setting the Object Manager SQL Log event parameter to 4 at the component event level. For more information about event logging, see *Event Logging Commands*.

For more information about Application Object Manager state values, statistics, and log files, about monitoring an Application Object Manager server component, and about event logging, see *Siebel System Monitoring and Diagnostics Guide* .

10 Siebel Server Components and Parameters

Siebel Server Components and Parameters

This chapter identifies the Siebel Server component groups and components and describes some of the server parameters that you might have to set. It includes the following topics:

- *Siebel Server Component Groups*
- *Siebel Server Components*
- *Siebel Enterprise, Server, and Component Parameters*

Siebel Server Component Groups

The following table lists some predefined Siebel Server component groups and the components they include. Many of the components, such as Application Object Manager components, are language-specific, although in most cases the full language-specific names are not shown in this table.

Component Group Name	Alias	Component Name	Alias
Assignment Management	AsgnMgmt	Batch Assignment	AsgnBatch
		Assignment Manager	AsgnSrvr
Auxiliary System Management	SystemAux	File System Manager	FSMSrvr
		Server Request Processor	SRProc
		Server Tables Cleanup	SvrTblCleanup
		Server Task Persistance	SvrTaskPersist
		Siebel Administrator Notification Component	AdminNotify
Siebel Call Center	CallCenter	Call Center Object Manager	SCCObjMgr
		Self Service Object Manager	SServiceObjMgr
Communications Management	CommMgmt	Communications Inbound Processor	CommInboundProcessor

Component Group Name	Alias	Component Name	Alias
		Communications Inbound Receiver	CommInboundRcvr
		Communications Session Manager	CommSessionMgr
		Communications Configuration Manager	CommConfigMgr
		Communications Outbound Manager	CommOutboundMgr
		Email Manager	MailMgr
		Page Manager	PageMgr
		Smart Answer Manager	SmartAnswer
Content Center	ContCtr	Content Project Publish	ContProjPub
		Content Project Start	ContProjStart
Dun and Bradstreet	DandB	D&B Update Mgr (D&B)	DBNUpMgrDNB
		D&B Update Mgr (Multi-task)	DNBUpMgrMultiTask
		D&B Update Mgr (Siebel)	DNBUpMgrSieb
Data Quality	DataQual	Data Quality Manager	DQMgr
Enterprise Application Integration	EAI	Custom Application Object Manager	CustomAppObjMgr
		Enterprise Integration Mgr	EIM
		EAI Object Manager	EAIObjMgr
		JMS Receiver	JMSReceiver
		MQSeries Server Receiver	MqSeriesSrvRcvr
		MQSeries AMI Receiver	MqSeriesAMIRcvr
		MSMQ Receiver	MSMQRcvr

Component Group Name	Alias	Component Name	Alias
Siebel eChannel	eChannel	Partner Manager Object Manager	PManagerObjMgr
		Siebel Partner Portal Object Manager	PartnerPortalObjMgr
		eChannel Object Manager	eChannelObjMgr
Siebel eDocuments	eDocuments	Document Server	DocServer
Forecast Service Management	FcstSvc	Forecast Service Manager	FcstSvcMgr
Field Service	FieldSvc	Field Service Cycle Counting Engine	FSCyccnt
		Service Order Fulfillment Engine	FSFulfill
		Field Service Mobile Inventory Transaction Engine	FSInvTxn
		Service Order Part Locator Engine	FSLocate
		Preventive Maintenance Engine	FSPrevMnt
		Field Service Replenishment Engine	FSRepl
		Appointment Booking Engine	ApptBook
		Optimization Engine	Optimizer
		Field Service Object Manager	SFSObjMgr
Handheld Synchronization	HandheldSync	Sales Mobile Object Manager	SalesmObjMgr
		Service Mobile Object Manager	ServicemObjMgr
Handheld Synchronization SIA	HandheldSyncSIS	eCG Sales Mobile Object Manager	CGMObjMgr
		ePharma Mobile Object Manager	ePharmaMObjMgr
		FINS Mobile Object Manager	FINSMObjMgr
Siebel ISS	ISS	Siebel Product Configuration Object Manager	eProdCfgObjMgr

Component Group Name	Alias	Component Name	Alias
		eSales Object Manager	eSalesObjMgr
		eCustomer Object Manager	eCustomerObjMgr
Marketing Object Manager	MktgOM	Marketing Object Manager	SMObjMgr
		eMarketing Object Manager	eMarketObjMgr
		eEvents Object Manager	eEventsObjMgr
Marketing Server	MktgSrv	List Import Service Manager	ListImportSvcMgr
Disconnected Mobile Synchronization	MobileSync	BatchSync	BatchSync
		Database Extract	DbXtract
		Mobile Data Extraction	MobileDbXtract
		Parallel Database Extract	PDbXtract
		Transaction Processor	TxnProc
		Transaction Router	TxnRoute
PIM Server Integration Management	PIMSI	PIMSI Engine	PIMSIEng
		PIMSI Dispatcher	PIMSIDispatcher
Sales Hierarchy Service	SalesHierSvc	Sales Hierarchy Service Manager	SalesHierSvcMgr
Search Processing	Search	Search Data Processor	SearchDataProcessor
		Search Incremental Index Processor	SearchIncrementalIndexProcessor
Siebel Remote	Remote	Generate New Database	GenNewDb
		Replication Agent	RepAgent
		Synchronization Manager	SynchMgr

Component Group Name	Alias	Component Name	Alias
		Transaction Merger	TxnMerge
Siebel RTI	RTI	RTI Batch	RTIBatch
Siebel Sales	Sales	Sales Object Manager	SSEObjMgr
Siebel Anywhere	SiebAnywhere	Upgrade Kit Builder	UpgKitBldr
Siebel Web Tools	SiebelWebTools	Siebel Web Tools Object Manager	SWToolsObjMgr
System Management	System	Server Manager	ServerMgr
		Server Request Broker	SRBroker
		Siebel Connection Broker	SCBroker
		Siebel Server	SiebSrvr
		Siebel Server Scheduler	SrvrSched
Task UI	TaskUI	Task Log Cleanup	TaskLogCleanup
Territory Management	TerritoryMgmt	Minor Alignment - Territory Rules Merge Manager	MinTerrMergeMgr
		Major Alignment - Territory Rules Merge Manager	MajTerrMergeMgr
Workflow Management	Workflow	Generate Triggers	GenTrig
		Workflow Monitor Agent	WorkMon
		Workflow Process Batch Manager	WfProcBatchMgr
		Workflow Process Manager	WfProcMgr
		Workflow Action Agent	WorkActn
		Workflow Recovery Manager	WfRecvMgr

Siebel Server Components

The following table lists some of the predefined Siebel Server components. Many of the components, such as Application Object Manager components, are language-specific, although in most cases the full language-specific names are not shown in this table.

Component Name	Alias	Mode	Multithrea	Description
Appointment Booking Engine	ApptBook	Batch	Yes	Books appointments. For more information, see *Siebel Field Service Guide* .
Assignment Manager	AsgnSrvr	Batch	Yes	Automatic data assignment engine that assigns positions, employees, and organizations to objects. To run Interactive Assignment, the Server Request Processor component must also be running. For more information, see *Siebel Assignment Manager Administration Guide* .
Batch Assignment	AsgnBatch	Batch	No	Batch assigns positions, employees, and organizations to objects. For more information, see *Siebel Assignment Manager Administration Guide* .
BatchSync	BatchSync	Batch	No	Extracts user databases for Siebel Mobile disconnected, processes pending transactions, and applies transactions to the Siebel Server. For more information, see *Siebel Mobile Guide: Disconnected* .
Call Center Object Manager	SCCObjMgr	Interactive	Yes	Application Object Manager for Siebel Call Center.
Communications Configuration Manager	CommConfigMgr	Batch	Yes	Downloads and caches communications configuration. For more information, see *Siebel CTI Administration Guide* .
Communications Inbound Processor	CommInboundProcessor	Batch	Yes	Processes queued communication events. For more information, see *Siebel Email Administration Guide* .
Communications Inbound Receiver	CommInboundRcvr	Batch	Yes	Queues inbound communication events. For more information, see *Siebel Email Administration Guide* .
Communications Outbound Manager	CommOutboundMgr	Batch	Yes	Sends messages to recipients associated with business object instances. For more information, see *Siebel Email Administration Guide* .

Component Name	Alias	Mode	Multithrea	Description
Communications Session Manager	CommSessionMgr	Batch	Yes	Interacts with users for utilizing communications channels. For more information, see *Siebel CTI Administration Guide* .
Content Project Publish	ContProjPub	Batch	Yes	Publishes a content project.
Content Project Start	ContProjStart	Batch	Yes	Starts a content project.
Custom Application Object Manager	CustomAppObjMgr	Interactive	Yes	Siebel Custom Application Object Manager.
D&B Update Mgr (D&B)	DNBUpMgrDNB	Batch	No	Updates D&B tables with subscription data. For more information, see *Siebel Applications Administration Guide* .
D&B Update Mgr (Multi-task)	DNBUpMgrMultiTask	Batch	Yes	Creates multiple D&B Update Mgr (D&B) or D&B Update Mgr (Siebel) processes by sending asynchronous requests. For more information, see *Siebel Applications Administration Guide* .
D&B Update Mgr (Siebel)	DNBUpMgrSieb	Batch	No	Updates Siebel database tables with subscription data. For more information, see *Siebel Applications Administration Guide* .
Data Quality Manager	Dqmgr	Batch	Yes	Cleanses data and deduplicates records. For more information, see *Siebel Data Quality Administration Guide* .
Database Extract	DbXtract	Batch	No	Extracts visible data for a Siebel Remote client. For more information, see *Siebel Remote and Replication Manager Administration Guide* .
Document Server	DocServer	Batch	Yes	Generates documents. For more information, see *Siebel Applications Administration Guide* .
EAI Object Manager	EAIObjMgr	Interactive	Yes	Siebel EAI Object Manager. For more information, see *Overview: Siebel Enterprise Application Integration* .
eCG Sales Mobile Object Manager	CGMObjMgr	Interactive	Yes	eCG Sales Mobile Object Manager. For more information, see Siebel Mobile applications documentation on *Siebel Bookshelf* .

Component Name	Alias	Mode	Multithrea	Description
eChannel Object Manager	eChannelObjMgr	Interactive	Yes	Siebel eChannel Object Manager. For more information, see *Siebel Partner Relationship Management Administration Guide* .
eCustomer Object Manager	eCustomerObjMgr	Interactive	Yes	Siebel eCustomer Object Manager. For more information, see *Siebel eService Administration Guide for Siebel Open UI* .
eEvents Object Manager	eEventsObjMgr	Interactive	Yes	Siebel eEvents Object Manager. For more information, see *Siebel Events Management Guide* .
Email Manager	MailMgr	Background	No	Sends individual email messages. For more information, see *Siebel Business Process Framework: Workflow Guide* .
eMarketing Object Manager	eMarketObjMgr	Interactive	Yes	Siebel eMarketing Object Manager. For more information, see *Siebel Marketing Installation and Administration Guide* .
Enterprise Integration Mgr	EIM	Batch	No	Integrates enterprise data to and from other systems. For this component to run, the Server Request Processor component must also be running. For more information, see *Siebel Enterprise Integration Manager Administration Guide* .
ePharma Mobile Object Manager	ePharmaMObjMgr	Interactive	Yes	ePharma Mobile Object Manager. For more information, see Siebel Mobile applications documentation on *Siebel Bookshelf* .
eSales Object Manager	eSalesObjMgr	Interactive	Yes	Siebel eSales Object Manager. For more information, see *Siebel eSales Administration Guide* .
Field Service Cycle Counting Engine	FSCyccnt	Batch	Yes	Field Service Cycle Counting Engine. For this component to run, the Server Request Processor component must also be running. For more information, see *Siebel Field Service Guide* .
Field Service Mobile Inventory Transaction Engine	FSInvTxn	Batch	Yes	Field Service Mobile Inventory Transaction Engine. For this component to run, the Server Request Processor component must also be running. For more information, see *Siebel Field Service Guide* .
Field Service Object Manager	SFSObjMgr	Interactive	Yes	Siebel Field Service Object Manager. For this component to run, the Server Request Processor component must also be running. For more information, see *Siebel Field Service Guide* .

Component Name	Alias	Mode	Multithrea	Description
Field Service Replenishment Engine	FSRepl	Batch	Yes	Replenishes inventory locations. For this component to run, the Server Request Processor component must also be running. For more information, see *Siebel Field Service Guide* .
File System Manager	FSMSrvr	Batch	Yes	Manages the Siebel File System. For more information, see *Administering the Siebel File System*.
FINS Mobile Object Manager	FINSMObjMgr	Interactive	Yes	FINS Mobile Object Manager. For more information, see Siebel Mobile applications documentation on *Siebel Bookshelf* .
Forecast Service Manager	FcstSvcMgr	Batch	Yes	Executes forecast operations. For more information, see *Siebel Forecasting Guide* .
Generate New Database	GenNewDb	Batch	No	Generates a new Oracle Database SE2 database template file for Siebel Remote. For more information, see *Siebel Remote and Replication Manager Administration Guide* .
Generate Triggers	GenTrig	Batch	No	Generates triggers for Workflow Manager and Assignment Manager. For more information, see *Siebel Business Process Framework: Workflow Guide* .
JMS Receiver	JMSReceiver	Background	No	Preconfigured receiver for inbound JMS messages. For more information, see *Overview: Siebel Enterprise Application Integration* .
List Import Service Manager	ListImportSvcMgr	Batch	Yes	Loads lists of data into the Siebel database. For this component to run, the Server Request Processor component must also be running. For more information, see *Siebel Marketing Installation and Administration Guide* .
Marketing Object Manager	SMObjMgr	Interactive	Yes	Siebel Marketing Object Manager. For more information, see *Siebel Marketing Installation and Administration Guide* .
Major Alignment - Territory Rules Merge Manager	MajTerrMergeMgr	Batch	Yes	Merges staging rules with production rules for major alignment. For more information, see *Siebel Territory Management Guide* .
Minor Alignment - Territory Rules Merge Manager	MinTerrMergeMgr	Batch	Yes	Merges staging rules with production rules for minor alignment. For more information, see *Siebel Territory Management Guide* .

Component Name	Alias	Mode	Multithrea	Description
Mobile Data Extraction	MobileDbXtract	Batch	No	Extracts visible data for a Siebel Mobile disconnected client. For more information, see *Siebel Mobile Guide: Disconnected* .
MQSeries AMI Receiver	MqSeriesAMIRcvr	Background	No	Preconfigured receiver for inbound MQSeries AMI messages. For more information, see *Transports and Interfaces: Siebel Enterprise Application Integration* .
MQSeries Server Receiver	MqSeriesSrvRcvr	Background	No	Preconfigured receiver for inbound MQSeries server messages. For more information, see *Transports and Interfaces: Siebel Enterprise Application Integration* .
MSMQ Receiver	MSMQRcvr	Background	No	Preconfigured receiver for inbound MSMQ server messages. For more information, see *Transports and Interfaces: Siebel Enterprise Application Integration* .
Optimization Engine	Optimizer	Batch	Yes	Optimizes vehicle routing. For more information, see *Siebel Field Service Guide* .
Page Manager	PageMgr	Background	No	Sends pages generated by the Workflow Manager. For more information, see *Siebel Business Process Framework: Workflow Guide* .
Parallel Database Extract	PDbXtract	Batch	No	Extracts visible data for a Siebel Mobile disconnected client or for a Siebel Remote or Replication Manager client. For more information, see *Siebel Mobile Guide: Disconnected* or *Siebel Remote and Replication Manager Administration Guide* .
Partner Manager Object Manager	PManagerObjMgr	Interactive	Yes	Siebel Partner Manager Object Manager. For more information, see *Siebel Partner Relationship Management Administration Guide* .
Siebel Partner Portal Object Manager	PartnerPortalObjMgr	Interactive	Yes	Siebel Partner Portal Object Manager. For more information, see *Siebel Partner Relationship Management Administration Guide* .
PIMSI Engine	PIMSIEng	Batch	Yes	Executes real-time business processes. For more information, see *Siebel Server Sync Guide* .
PIMSI Dispatcher	PIMSIDispatcher	Batch	Yes	Executes real-time business processes. For more information, see *Siebel Server Sync Guide* .

Component Name	Alias	Mode	Multithrea	Description
Preventive Maintenance Engine	FSPrevMnt	Batch	Yes	Generates service requests and activities for preventive maintenance. For more information, see *Siebel Field Service Guide* .
Replication Agent	RepAgent	Background	No	Synchronizes a regional database with a parent database. For more information, see *Siebel Remote and Replication Manager Administration Guide* .
RTI Batch	RTIBatch	Batch	No	Executes SQL statements in a batch. For more information, see *Siebel Territory Management Guide* .
Sales Hierarchy Service Manager	SalesHierSvcMgr	Batch	Yes	Batch executes sales hierarchy service operations.
Sales Mobile Object Manager	SalesmObjMgr	Interactive	Yes	Siebel Sales Mobile Object Manager. For more information, see Siebel Mobile applications documentation on *Siebel Bookshelf* .
Sales Object Manager	SSEObjMgr	Interactive	Yes	Siebel Sales Object Manager.
Search Data Processor	SearchDataProcessor	Batch	Yes	Processes search data and builds indexes. For more information, see *Siebel Search Administration Guide* .
Search Incremental Index Processor	SearchIncrementalIndexProc	Batch	Yes	Processes search data and builds an index incrementally. For more information, see *Siebel Search Administration Guide* .
Self Service Object Manager	SServiceObjMgr	Interactive	Yes	Siebel Self Service Object Manager. For more information, see *Siebel eService Administration Guide for Siebel Open UI* .
Server Manager	ServerMgr	Interactive	No	Administers configuration data within the Siebel Enterprise Server. For more information, see *About Siebel Server Manager*.
Server Request Broker	SRBroker	Interactive	Yes	Routes requests and asynchronous notifications among clients and components. For more information, see *About Server Request Broker (SRBroker)*.
Server Request Processor	SRProc	Background	Yes	Server request scheduler and request or notification store-and-forward processor. For more information, see *About Server Request Processor (SRProc)*.

Component Name	Alias	Mode	Multithrea	Description
Server Tables Cleanup	SvrTblCleanup	Background	No	Deletes completed and expired server request records. For more information, see *About Server Tables Cleanup (SvrTblCleanup)*.
Server Task Persistance	SvrTaskPersist	Background	No	Persists all of the tasks created by the Siebel Server.
Service Mobile Object Manager	ServicemObjMgr	Interactive	Yes	Siebel Service Mobile Object Manager. For more information, see Siebel Mobile applications documentation on *Siebel Bookshelf* .
Service Order Fulfillment Engine	FSFulfill	Batch	Yes	Fulfills pending service orders. For more information, see *Siebel Field Service Guide* .
Service Order Part Locator Engine	FSLocate	Batch	Yes	Locates pending service orders. For more information, see *Siebel Field Service Guide* .
Siebel Administrator Notification Component	AdminNotify	Batch	Yes	Administers the server component notification feature. For more information, see *About System Alert Notification*.
Siebel Connection Broker	SCBroker	Background	No	Brokers inbound connection requests for interactive mode server components (Application Object Managers) and load-balances session requests between multiple instances of interactive mode server components. For more information, see *About Siebel Connection Broker (SCBroker)*.
Siebel Product Configuration Object Manager	eProdCfgObjMgr	Interactive	Yes	Configuration server for complex products. For more information, see *Siebel Product Administration Guide* .
Siebel Server	SiebSrvr	Background	No	Siebel Server root process and network listener.
Siebel Server Scheduler	SrvrSched	Background	No	Supports the running of Siebel Server and server components by spawning component processes as requested by the Siebel Server. For more information, see *About Siebel Server Scheduler (SrvrSched)*.
Siebel Web Tools Object Manager	SWToolsObjMgr	Interactive	Yes	Siebel Web Tools Object Manager. For more information, see *Using Siebel Tools* .
Smart Answer Manager	SmartAnswer	Batch	No	Categorizes text in email messages. For more information, see *Siebel Smart Answer Guide* .

Component Name	Alias	Mode	Multithrea	Description
Synchronization Manager	SynchMgr	Interactive	Yes	Manages Siebel Remote and Replication Manager synchronization sessions. For more information, see *Siebel Remote and Replication Manager Administration Guide* .
Task Log Cleanup	TaskLogCleanup	Background	No	Cleans up the task transaction storage after transactions are committed. For more information, see *Siebel Business Process Framework: Task UI Guide* .
Transaction Merger	TxnMerge	Background	No	Merges transactions from Siebel Remote and Replication Manager clients into the Siebel database. For more information, see *Siebel Remote and Replication Manager Administration Guide* .
Transaction Processor	TxnProc	Background	No	Prepares the transaction log for the Transaction Router. For more information, see *Siebel Mobile Guide: Disconnected* or *Siebel Remote and Replication Manager Administration Guide* .
Transaction Router	TxnRoute	Background	No	Routes visible transactions to Siebel Mobile disconnected clients or to Siebel Remote or Replication Manager clients. For more information, see *Siebel Mobile Guide: Disconnected* or *Siebel Remote and Replication Manager Administration Guide* .
Upgrade Kit Builder	UpgKitBldr	Batch	Yes	Creates the upgrade kit based on information collected by the Upgrade Kit Wizard. For more information, see *Siebel Anywhere Administration Guide* .
Workflow Action Agent	WorkActn	Background	No	Executes Workflow Manager actions. For more information, see *Siebel Business Process Framework: Workflow Guide* .
Workflow Monitor Agent	WorkMon	Background	No	Monitors Workflow Manager events. For more information, see *Siebel Business Process Framework: Workflow Guide* .
Workflow Process Batch Manager	WfProcBatchMgr	Batch	Yes	Executes workflow processes in a batch. For more information, see *Siebel Business Process Framework: Workflow Guide* .
Workflow Process Manager	WfProcMgr	Batch	Yes	Executes real-time workflow processes. For more information, see *Siebel Business Process Framework: Workflow Guide* .
Workflow Recovery Manager	WfRecvMgr	Batch	Yes	Recovers interrupted workflow processes due to Siebel Server failures. For more

Component Name	Alias	Mode	Multithrea	Description
				information, see *Siebel Business Process Framework: Workflow Guide* .

Siebel Enterprise, Server, and Component Parameters

The table that follows lists some of the Siebel Enterprise Server, Siebel Server, and generic component parameters and their related attributes.

More complete descriptions of each listed parameter follow this table. In most cases, the parameters with *** shown as the default value are initially set during the initial configuration tasks (based on the input specified by the administrator). For the parameters with +++ shown as the default value, the actual default values differ from component to component. Review the documentation pertinent to the individual component for more information.

This topic contains the following topics:

- *Siebel Enterprise Server Parameters*
- *Siebel Server Parameters*
- *Generic Parameters*

Note: Not all of the parameters used by Siebel CRM applications are described in this book. Many parameters are primarily described elsewhere on *Siebel Bookshelf* , while some parameters are not documented on *Siebel Bookshelf* .

Parameter Name	Alias	Level	Req	Override	Effective Immed	Dynamic	Default Value
Enterprise Server Description	EnterpriseDesc	Enterprise	No	No	No	No	***
Indexspace Name	IdxSpace	Enterprise	No	Yes	Yes	Yes	***
ODBC Data Source	Connect	Enterprise	Yes	Yes	Yes	No	***
Siebel File System	FileSystem	Enterprise	Yes	Yes	Yes	No	***
Siebel Repository	Repository	Enterprise	No	Yes	No	No	Siebel Repository
Table Owner	TableOwner	Enterprise	No	No	No	No	***
Table Owner Password	TableOwnPass	Enterprise	No	No	No	Yes	***
Tablespace Name	TblSpace	Enterprise	No	No	No	No	***

Parameter Name	Alias	Level	Req	Override	Effective Immed	Dynamic	Default Value
Upgrade Component	UpgradeComponent	Enterprise	No	No	Yes	No	Siebel HQ Server
Auto Startup Mode	AutoStart	Server	No	No	Yes	No	True
Communication Transport	Comm	Server	Yes	No	Yes	No	TCPIP
Communication Type	CommType	Server	Yes	No	Yes	No	NONE
Compression Type	Compress	Server	No	No	Yes	No	NONE
Component Priority Level Timeout	CompPriorityTime	Server	No	No	No	No	Not applicable
Encryption Type	Crypt	Server	No	No	Yes	No	NONE
Host Name	Host	Server	Yes	No	No	No	***
Log Archive Keep	LogArchive	Server	No	No	Yes	No	10
Log Segment Size	LogSegmentSize	Server	No	No	No	No	Not applicable
Log Maximum Segments	LogMaxSegments	Server	No	No	No	No	Not applicable
Server Description	ServerDesc	Server	No	No	No	No	***
Server Shutdown Wait Time	ShutdownTime	Server	No	No	Yes	Yes	60
Siebel Root Directory	RootDir	Server	Yes	No	No	No	***
Siebel Server Name	Server	Server	Yes	No	Yes	No	***
Size of Error Buffer	ErrorBufferSize	Server	No	No	No	No	1000
Synchronization Port	SyncPort	Server	Yes	No	Yes	No	40400
Alert Level	AlertLevel	Generic	No	Yes	Yes	Yes	1

Parameter Name	Alias	Level	Req	Override	Effective Immed	Dynamic	Default Value
Application Datasource	CFGDatasource	Generic	No	Not applicable	Yes	No	ServerDataSrc
Application Scripting Enabled	CFGEnableScripting	Generic	No	Not applicable	Yes	No	True
Application Shared Mode users directory	CFGSharedModeUsersDir	Generic	No	Not applicable	Yes	No	Not applicable
Auto Restart	AutoRestart	Generic	No	No	No	No	False
Compressed File Download	CompressedFileDownload	Generic	No	No	No	No	True
DB Multiplex - Max Number of Shared DB Connections	MaxSharedDbConns	Generic	No	Not applicable	No	No	-1
DB Multiplex - Min Number of Dedicated DB Connections	MinTrxDbConns	Generic	No	Not applicable	No	No	-1
DB Multiplex - Min Number of Shared DB Connections	MinSharedDbConns	Generic	No	Not applicable	No	No	-1
Default Tasks	DfltTasks	Generic	Yes	No	No	No	0
Error Flags	ErrorFlags	Generic	No	Yes	Yes	Yes	0
Honor MaxTasks	HonorMaxTasks	Generic	No	No	No	No	False
Language Code	Lang	Generic	Yes	Yes	Yes	No	ENU
Locale Code	LocaleCode	Generic	Yes	Yes	Yes	No	***
Local load balancing upper threshold	UpperThreshold	Generic	No	Yes	No	No	100
Log Print Timestamp	LogTimestamp	Generic	No	Yes	Yes	Yes	False

Parameter Name	Alias	Level	Req	Override	Effective Immed	Dynamic	Default Value
Maximum depth of the query	MaxDepthOfTheQuery	Generic	No	No	No	No	500
Maximum MT Servers	MaxMTServers	Generic	Yes	No	No	No	1
Maximum Processes	MaxProcs	Generic	Yes	Yes	No	No	20
Maximum Tasks	MaxTasks	Generic	Yes	No	No	No	+++
Memory Usage Based Multithread Shell Recycling	MemoryBasedRecycle	Generic	No	Yes	No	No	False
Minimum MT Servers	MinMTServers	Generic	Yes	No	No	No	1
Minimum Up Time	MinUpTime	Generic	No	No	No	No	60
Multithreaded	Threaded	Generic	Yes	No	No	No	False
Number of lines after which to flush the log file	LogFlushFreq	Generic	No	No	Yes	No	0
Number of Restarts	NumRestart	Generic	No	No	No	No	10
Number of Retries	NumRetries	Generic	No	No	No	No	10000
Number of Sessions per SISNAPI Connection	SessPerSisnConn	Generic	No	Not applicable	No	No	20
OM - Data Source	DataSource	Generic	No	Not applicable	Yes	No	ServerDataSrc
OM - Named Data Source name	NamedDataSource	Generic	No	Not applicable	Yes	No	ServerDataSrc, GatewayDataSrc
OM - Model Cache Maximum	ModelCacheMax	Generic	No	Not applicable	Yes	No	10

Parameter Name	Alias	Level	Req	Override	Effective Immed	Dynamic	Default Value
OM - Preload Links for batch components	PreloadLinksForBatchComp	Generic	No	Not applicable	No	Yes	True
OM - Resource Language Code	ResourceLanguage	Generic	No	Not applicable	Yes	No	Not applicable
OM - Save Preferences	SavePreferences	Generic	No	Yes	No	No	True
Oracle Degree of Parallelism	OraDegreeOfParallelism	Generic	No	Not applicable	No	No	auto
Oracle Optimizer Ignore Hints	OraOptimizerIgnoreHints	Generic	No	Not applicable	No	No	False
Password	Password	Generic	Yes	Yes	Yes	Yes	***
Process VM Usage Lower Limit	MemoryLimit	Generic	No	Yes	No	No	1500
Process VM Usage Upper Limit	MemoryLimitPercent	Generic	No	Yes	No	No	20
Retry Interval	RetryInterval	Generic	No	No	No	No	5
Retry Up Time	RetryUpTime	Generic	No	No	No	No	600
Session Keepalive Timeout	SessKeepAlive	Generic	No	Yes	No	No	7200
SISNAPI Connection Maximum Idle Time	ConnIdleTime	Generic	No	Yes	Yes	No	-1
SISNAPI - Log Traffic	LogTraffic	Generic	No	Not applicable	Yes	Yes	False
Sleep Time	SleepTime	Generic	No	Yes	Yes	Yes	60
SQL Trace Flags	SQLFlags	Generic	No	Yes	Yes	Yes	0
Static Port Number	PortNumber	Generic	Yes	No	Yes	No	0
Trace Flags	TraceFlags	Generic	No	Yes	Yes	Yes	0

Parameter Name	Alias	Level	Req	Override	Effective Immed	Dynamic	Default Value
Use IP Address	UseIPAddress	Generic	No	No	No	No	True
Use Shared Log File	LogUseSharedFile	Generic	No	Not applicable	No	No	False
User Name	Username	Generic	Yes	Yes	Yes	Yes	***

Siebel Enterprise Server Parameters

This topic describes in detail the Siebel Enterprise Server parameters listed in *Siebel Enterprise, Server, and Component Parameters*.

Enterprise Server Description (alias EnterpriseDesc). A description of the Siebel Enterprise Server, used for identification in Server Manager views. The system prompts you for the value of this parameter during the configuration of the Siebel Enterprise Server after installation.

Indexspace Name (alias IdxSpace). The indexspace name for the Siebel database schema tables. This parameter specifies the name of the storage space in which to create the indexes for the Siebel database schema. The exact physical meaning of this parameter is database platform-specific. To determine whether this parameter applies to your database platform, see *Siebel Installation Guide* .

ODBC Data Source (alias Connect). The ODBC data source name for the Siebel database schema connectivity. The default data source is created during the initial configuration of the Siebel Server, but can be overridden for a component or task. This data source must be created as a system DSN. Note that this parameter is case-sensitive.

Siebel File System (alias FileSystem). The Siebel File System path names. The specification of the File System paths must be valid relative to the computer on which the Siebel Server is installed. The value of the parameter cannot exceed 2048 characters. Multiple File System directories can be specified for this parameter by using commas to delimit each directory. For example, the following value specifies two directories hosted on different server computers running Microsoft Windows:

```
\\server_name1\fs\,\\server_name2\fs\
```

Siebel Repository (alias Repository). The name of the Siebel Repository for application configuration information. The default value is Siebel Repository. There can only be one active repository for each database.

Table Owner (alias TableOwner). The table owner for the Siebel database schema:

- For Oracle Database, you are prompted for the default value for Table Owner during the Siebel Server initial configuration.
- For Microsoft SQL Server, the value defaults to dbo.

Table Owner Password (alias TableOwnPass). The database password for the table owner account. This value must be set in order to run Siebel Server components that manipulate objects in the Siebel database schema.

Tablespace Name (alias TblSpace). The tablespace name for the Siebel database schema tables. This parameter specifies the name of the storage space in which to create the tables for the Siebel database schema. The exact physical meaning of this parameter is database platform-specific. To determine whether this parameter applies to your database platform, see *Siebel Installation Guide* .

Upgrade Component (alias UpgradeComponent). Used by Siebel Anywhere to determine which Siebel Anywhere configuration is version-checked. By default, the value is Siebel HQ Server. On a regional Siebel Server, change this value to Siebel Regional Server.

Siebel Server Parameters

This topic describes in detail the Siebel Server parameters listed in *Siebel Enterprise, Server, and Component Parameters*.

Auto Startup Mode (alias AutoStart). Indicates whether the Siebel Server components start automatically on Siebel Server startup. This parameter defaults to True, which indicates that the Siebel Server components are fully enabled and the default number of Siebel Server processes start when the Siebel Server system service starts (or the computer restarts). If Auto Startup Mode is set to False, then the Siebel Server components enter a shutdown state after the Siebel Server system service starts.

Communication Transport (alias Comm). The name of the transport type for network communications, for example, TCPIP.

Communication Type (alias CommType). The name of the communication type for network communications, for example, TLS or NONE. See also *Siebel Remote and Replication Manager Administration Guide* and *Siebel Security Guide* .

Component Priority Level Timeout (alias CompPriorityTime). The amount of time to wait before starting lower-priority components. Components are prioritized as follows:

- Components in the System Management component group have the highest priority. These components start first.
- Components in the Auxiliary System Management component group start next.
- Components in the remaining component groups have the lowest priority.

The Siebel Server starts components in the System Management component group and waits for the maximum number of seconds, specified by the Component Priority Level Timeout parameter, for these components to initialize. If, at the expiry of Component Priority Level Timeout, the components fail to initialize, then the Siebel Server attempts to start the components in the Auxiliary System Management component group. If the components in this component group fail to initialize before the expiry of Component Priority Level Timeout, then the Siebel Server attempts to start the lowest priority components.

> **Note:** The Siebel Server attempts to start lower-level components irrespective of the dependencies of these components in the Auxiliary System Management component group, such as File System Manager or Server Request Processor.

Compression Type (alias Compress). The type of compression (NONE or ZLIB) for network communications sent internally between the Application Object Manager and the Siebel Application Interface. See also the information about configuring Siebel Application Interface profiles using Siebel Management Console and about the setting HTTP 1.1-Compliant Firewall / Enable Web Compression, in *Siebel Installation Guide* .

Encryption Type (alias Crypt). The type of encryption for network communications between Siebel modules such as Siebel Application Interface and the Application Object Manager. See also the information about configuring Siebel Server profiles using Siebel Management Console and about the setting Security Encryption Level or Type, in *Siebel Installation Guide* . As of Siebel CRM 19.6 Update, you can use Transport Layer Security (TLS) for Siebel Remote, and you cannot use RSA. If Crypt is set to RSA for the Synchronization Manager, then you must change this value to NONE. See

the description of the CommType parameter and see *Siebel Remote and Replication Manager Administration Guide* . For more information about encryption, see *Siebel Security Guide* .

Host Name (alias Host). The name of the host computer on which the Siebel Server is installed. The value is set automatically during the initial configuration of the Siebel Server, but can be changed if you want to route connection requests through a network card bound to a different host name.

Log Archive Keep (alias LogArchive). The number of log archive directories to keep in the `logarchive` directory. Each time the Siebel Server system service starts, the current `log` subdirectory moves to the `logarchive` subdirectory, tagged with the incarnation number of the Siebel Server. This parameter indicates the number of previous `logarchive` directories to retain. If this parameter is set to 0, then the current `log` subdirectory is not archived upon startup of the Siebel Server system service. If this parameter is set to -1, then the Siebel Server keeps `logarchive` subdirectories. After moving the `log` directory, a new `log` directory is created, inheriting the permissions from the parent `siebsrvr` directory.

Log Segment Size (alias LogSegmentSize). Determines how large a segment of the log file is, in 512 byte blocks. For example, if you set this parameter to 10, then the log segment will be 5 KB in size. A value of 0 turns off segmentation.

Log Maximum Segments (alias LogMaxSegments). Determines the number of log segments that are kept. If you set this parameter to 1, then only one segment is kept. When this segment reaches the size specified by the Log Segment Size parameter, the segment is overwritten. In general, set this parameter to a high value, such as 20. In this case, the twenty-first segment would overwrite the first segment, and so forth.

Server Description (alias ServerDesc). A description of the Siebel Server, used for identification in Siebel Server Manager views. The value of this parameter is prompted for during the initial configuration of the Siebel Server.

Server Shutdown Wait Time (alias ShutdownTime). The time to wait (in seconds) during a Siebel Server shutdown before killing component processes. When a Siebel Server is shut down (either from the Siebel Server Manager, when the Siebel Server system service is stopped, or when the computer is shut down or restarted), the currently running component tasks are notified. If the tasks do not shut down within the time specified by the Server Shutdown Wait Time parameter, then the Siebel Server kills the component processes directly and then finishes shutting down. The default value of this parameter is 60 seconds.

Siebel Root Directory (alias RootDir). The root installation directory for the Siebel Server. Every Siebel Server subdirectory is directly under this directory (such as `admin`, `dbtempl`, `docking`, `log`, `logarchive`, `upgrade`, and so on). Do not change the value for this parameter.

Siebel Server Name (alias Server). The name of the Siebel Server. This parameter is specified during the initial configuration of the Siebel Server. The name of the Siebel Server cannot be changed after it is initially configured. The Siebel Server Name parameter can contain only alpha characters, numerals, underscores, or a combination thereof. Parameter names must also lead with an alpha character and can be no longer than 12 characters. For more information about the name of the Siebel Server, see *Siebel Installation Guide* .

Size of Error Buffer (alias ErrorBufferSize). The number of entries in the shared memory circular error buffer used by the Siebel Server. This buffer receives all level 0 and level 1 error messages.

Synchronization Port (alias SyncPort). The TCP/IP port number for the Synchronization Server component. The Mobile Web Clients that synchronize with this Siebel Server must be configured to connect to this port when initiating a synchronization session (in the DockConnString parameter of the client configuration file).

Generic Parameters

This topic describes in detail some of the generic parameters listed in *Siebel Enterprise, Server, and Component Parameters*. Many of these parameters can be set at the Siebel Enterprise Server, Siebel Server, or component levels.

Alert Level (alias AlertLevel). The level of logging to the Server Alert File. The value is set to 1 by default, but more detailed information can be specified by setting the parameter to a higher value. Currently, only levels 1 and 2 are supported. Level 1 sends only information about abnormal process and task terminations to the Alert File. Level 2 sends information about every process or task when it exits, whether normally or unexpectedly.

Application Datasource (alias CFGDatasource). The Application Object Manager default configuration data source. The value for this parameter is one of the values listed for the OM - Named Data Source name parameter.

Application Scripting Enabled (alias CFGEnableScripting). Determines whether the Application Object Manager server component can execute a server script. If you set this to True, then scripting DLL files are loaded and the application can then execute scripts. The default value for this parameter is True. Setting this parameter to False also disables browser scripts in addition to server scripts.

Application Shared Mode users directory (alias CFGSharedModeUsersDir). Specifies the users directory used in shared mode. Update the value of this parameter if you change the location of the Siebel File System.

Application * Font. Several parameters with names starting with *Application* and ending in *Font* specify fonts to use for Siebel application user interface elements unless these settings are overridden by font specifications in style sheets (CSS).

Auto Restart (alias AutoRestart). Indicates whether this component can be restarted automatically. This parameter works in conjunction with the Number of Restarts parameter to determine the number of attempts that are made to restart the target component.

> **Note:** For multithreaded server components, auto-restart does not occur if a process initially fails to start and is terminated by the Siebel application. Only processes that reach the running state are restarted.

Compressed File Download (alias CompressedFileDownload). By default, files are downloaded to a client's browser in a compressed form when using Siebel File System Manager. On certain versions of Microsoft Internet Explorer, this operation might result in the renaming of the file. If you want to disable compressed file download, then set this parameter to False for the Application Object Manager for which you want this feature to be disabled. The default value is True.

DB Multiplex - Max Number of Shared DB Connections (alias MaxSharedDbConns). One of two parameters that configure shared database connections. (The other parameter is DB Multiplex - Min Number of Shared DB Connections.) Shared connections are used by most Application Object Manager operations. DB Multiplex - Max Number of Shared DB Connections controls the maximum number of shared database connections, and is defined for each component. That is, DB Multiplex - Max Number of Shared DB Connections controls the maximum total number of shared database connections for the component on each Siebel Server, not an instance (task or process) of the component. A setting of -1 disables this parameter and is the default setting.

DB Multiplex - Min Number of Dedicated DB Connections (alias MinTrxDbConns). Controls the minimum number of dedicated database connections within an Application Object Manager process. Dedicated database connections are used primarily by specialized Siebel components, such as Siebel EAI, that need transactions to span multiple Application Object Manager operations. The DB Multiplex - Min Number of Dedicated DB Connections parameter is defined for each instance of the component. That is, DB Multiplex - Min Number of Dedicated DB Connections controls the minimum number of dedicated database connections for each instance (process) of the component, not for the entire component. This functionality is different from the parameters that configure shared database connections. A setting of -1 disables this parameter and is the default setting.

DB Multiplex - Min Number of Shared DB Connections (alias MinSharedDbConns). One of two parameters that configure shared database connections. (The other parameter is DB Multiplex - Max Number of Shared DB Connections.) Shared connections are used by most Application Object Manager operations. DB Multiplex - Min Number of Shared DB Connections controls the minimum number of shared database connections, and is defined

for each component. That is, DB Multiplex - Min Number of Shared DB Connections controls the minimum number of shared database connections a component tries to maintain on each Siebel Server across all instances of this component. A setting of -1 disables this parameter and is the default setting.

> **Note:** Set the DB Multiplex - Min Number of Shared DB Connections parameter less than the value of the Maximum Tasks (alias MaxTasks) parameter. Setting this value greater than MaxTasks disables database connection pooling.

Default Tasks (alias DfltTasks). The number of processes to start for a background mode component when the component is started explicitly through the Siebel Server Manager, or when the Siebel Server is started (if the component state was last set to Running). Components with a Default Tasks parameter set to a value greater than zero start automatically when the Siebel Server is started.

DefaultNavigation (alias DefaultNavigation). The default navigation control in the Siebel Web Client application, which can be NAVIGATION_SIDE, NAVIGATION_TAB, or NAVIGATION_TREE. The default setting is NAVIGATION_SIDE, which means that the application uses the Side Menu navigation control by default. The navigation control user preference, if set, takes precedence over the DefaultNavigation component parameter value. For Siebel Web Tools, Side Menu navigation is always enabled regardless of the parameter or user preference settings. This parameter was added in Siebel CRM 19.4 Update. For more information, see *Siebel Fundamentals Guide* and *Siebel Applications Administration Guide* .

EnabledTC (alias EnabledTC). When set to True for an Application Object Manager, enables a terms and conditions page that users must accept to be allowed into the Siebel application. Additional configuration steps apply, which are described on My Oracle Support in article 478253.1 (Article ID).

EnableMultiTab (alias EnableMultiTab). This parameter enables users to access the same applications over multiple tabs of the same browser session. You can also limit the number of tabs in which a user can access the Siebel CRM applications, by setting the Maximum Possible Tabbed Sessions value when you configure the Siebel Application Interface profile in the Siebel Management Console. This parameter was added in Siebel CRM 19.7 Update. For more information, see *Configuring Siebel Open UI* . See also *Siebel Installation Guide* .

Enable Safe Boot (alias EnableSafeBoot). The out-of-the-box (OOTB) Siebel Runtime Repository is shipped as a loadable unit (DLL or SO file). When the EnableSafeBoot parameter is set to True, the application uses the OOTB Runtime Repository. You can use this parameter in testing or troubleshooting of your custom configurations. For Siebel Web Tools, SafeBoot is always enabled regardless of the parameter setting. This parameter was added in Siebel CRM 17.0. For more information, see *Configuring Siebel Open UI* .

Error Flags (alias ErrorFlags). Specifies the flags used for tracing error information. This parameter is used to turn on various types of component-specific error tracing.

Export Status Size (alias ExportStatusSize). Specifies the number of records to be exported in an export operation performed by the Application Object Manager component.

Honor MaxTasks (alias HonorMaxTasks). When the parameter value is True, a component process that reaches Max Tasks stops accepting requests from the Server Request Broker. If another request is sent, then an error message results. The process resumes accepting requests after some tasks finish. If the parameter value is False, then all of the requests are queued in the component process.

Language Code (alias Lang). The three-letter language code for the component processes. Translatable messages (including error messages) are output in the specified language. The translated message files for the language must exist in the locale subdirectory of the Siebel Server installation. For more information, see *Siebel Global Deployment Guide* .

Locale Code (alias LocaleCode). The three-letter locale code for the component processes. A locale is a set of rules guiding how common data is displayed to the user or is received from the user. Siebel CRM supports formatting of data,

such as dates, time, numbers, and currency, based on locale settings. Locales are administered using the Locale view in the Administration - Data screen. For more information, see *Siebel Global Deployment Guide* .

> **Note:** Siebel CRM uses the three-letter code conventions of Microsoft for locale and language code. For more information about setting locales, see *Siebel Applications Administration Guide* .

Local load balancing upper threshold (alias UpperThreshold). Determines the load threshold at which a new Application Object Manager process starts, and is based on the load percentage of a given Application Object Manager process. For example, setting this parameter to 80 percent on an Application Object Manager that handles 50 concurrent sessions starts a new Application Object Manager process when 40 sessions become active. The default value is 100 percent. For more information about load balancing, see *Siebel Deployment Planning Guide* .

> **Note:** If the Application Object Manager parameter Minimum MT Servers is equal to Maximum MT Servers, then the effective value for this parameter is 100 percent, despite the actual setting.

Log Print Timestamp (alias LogTimestamp). Specifies whether to print a timestamp on records written to the trace files. The value is set to True by default, but administrators might want to override it to False for components that perform a large amount of logging (or if a high value is set for the Trace Flags or SQL Trace Flags parameters).

Manifest Safe Load (alias ManifestSafeLoad). Set to True to start the application with only the Oracle-provided (out of the box, or OOTB) manifest records. When False (the default), the application starts with both customer (custom-configured) and seeded (Oracle-provided) manifest data. For Siebel Web Tools, the OOTB manifest is always used regardless of the parameter setting. This parameter was added in Siebel CRM 19.4 Update. For more information, see *Configuring Siebel Open UI* .

Maximum depth of the query (alias MaxDepthOfTheQuery). Allows you to increase the maximum parse depth of a query with a large search specification, including, for example, a large number of OR or AND clauses.

You might receive this error: `The query could not be run because it is too big`. If you receive errors indicating that a query cannot run because it is too big, then you might refine the query to be more selective or to use parentheses. Alternatively, you can try setting this parameter to a larger value to eliminate the error. The default value is 500.

If you need to adjust this parameter value, then the setting you choose must correlate to the default stack size for your server operating system. For example, a value of 800 will generally work on Microsoft Windows, for which the default stack size is 1 MB, while a value of 500 will generally work on Linux or UNIX operating systems, for which the default stack size is 512 KB. If you increase the stack size for your operating system, then you can use a higher setting for this parameter.

You can optionally set this parameter at the enterprise level or the server level. You can also set MaxDepthOfTheQuery in the application configuration file for the Siebel Mobile Web Client or Developer Web Client (in the `[InfraObjMgr]` section), or set it in the tools.cfg file for Siebel Tools.

> **CAUTION:** Changing the operating system stack size can adversely impact the computer or application performance and should not be attempted without due diligence and consultation with experts.

Maximum MT Servers (alias MaxMTServers). Specifies the maximum number of multithreaded Siebel Server processes to be run concurrently for a component that has multithreading enabled. Note that only batch mode and interactive mode components can run with multithreaded set to True. Tasks and client sessions run as threads within the multithreaded Siebel Server processes.

The number of tasks that can run in each Siebel Server process is determined by the value of the Maximum Tasks parameter divided by Maximum MT Servers. Increase or decrease the value of Maximum MT Servers based on the

number of users for the given component process. For more information about this parameter, see *Siebel Performance Tuning Guide* .

Maximum Processes (alias MaxProcs). Specifies the maximum number of concurrent running processes for a Siebel Server component. The Siebel Server must be restarted in order for any changes to this parameter to take effect.

Maximum Tasks (alias MaxTasks). Specifies the maximum number of background mode, batch mode, or interactive mode processes or threads that can run concurrently for a component. This value applies to threads for components that have multithreading enabled, or otherwise component processes. Increase or decrease this value based on the number of users for the given component process. This value also determines the number of tasks for each component that are tracked by the Siebel Server. For more information about this parameter, see *Siebel Performance Tuning Guide* .

> **Note:** Multithreaded Siebel Server processes are not included in the counting of tasks. The tasks run as threads within the processes. These multithreaded processes are guided by the Minimum MT Servers and Maximum MT Servers parameters.

Memory Usage Based Multithread Shell Recycling (alias MemoryBasedRecycle). If it is set to True, then processes for this component are recycled automatically when virtual memory usage reaches a specified threshold. The threshold is set using the parameter Process VM Usage Lower Limit. For example, when a component is set with this parameter and the memory usage has exceeded the configured threshold, the recycling procedure begins by disabling new tasks, spawning a new process, and commencing a normal shutdown (that is, waiting for all of the tasks to finish before shutting down). The default value is False. Use this parameter to remedy your application only if excessive memory usage created by memory leaks seems to be occurring.

Minimum MT Servers (alias MinMTServers). Specifies the default number of multithreaded Siebel Server processes that are started for a component that has multithreading enabled. These processes are brought up when the component is started explicitly through the Siebel Server Manager, or when the Siebel Server is started (if the component state was last set to Running).

Additional multithreaded Siebel Server processes are started as needed (namely, when the maximum number of threads that can run in a Siebel Server process has been reached), up to the value of the Maximum MT Servers parameter. Setting this parameter to 0 disables the component. For more information about this parameter, see *Siebel Performance Tuning Guide* .

Minimum Up Time (alias MinUpTime). Specifies the minimum time that a multithreaded server component such as an Application Object Manager must be up for a restart to be successful (in seconds). In order for the component restart to be considered successful, the component must be running for the duration specified by this parameter. This parameter works with the Number of Restarts parameter to determine the number of restart attempts in a time interval allowed for a component (NumRestart multiplied by MinUpTime). If a component instance cannot be restarted after this time interval, then no new restart is attempted (the component instance will not be running). The default value for this parameter is 60 seconds.

Multithreaded (alias Threaded). Specifies whether the component is multithreaded or multiprocess. This parameter only applies to batch mode and interactive mode components. Use the default value for this parameter.

> **CAUTION:** Do not change the value of this parameter without direct guidance Global Customer Support. For help with changing the value of this parameter, create a service request (SR) on My Oracle Support. Alternatively, you can phone 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.

Number of lines after which to flush the log file (alias LogFlushFreq). Specifies how often data is written to the log file. Set the value to *n* so that data is written to the log file every *n* lines. For example, set the value to 1 to write every line

to the log file. Set the value to 0 (default value) to allow the operating system to determine when to write data to the log file.

Number of Restarts (alias NumRestart). Specifies the number of times that a multithreaded server component can be restarted if it exited with errors in less than the time set for Minimum Up Time. This parameter works with Auto Restart to determine whether multithreaded server components will be restarted. This parameter also works with the Minimum Up Time parameter to determine the number of restart attempts in a time interval allowed for a component (NumRestart multiplied by MinUpTime). If a component instance cannot be restarted after this time interval, then no new restarts are attempted. (In this situation, the component instance will not be running.) The default value for this parameter is 10.

Number of Retries (alias NumRetries). Specifies the number of retries for recovery. This parameter works with the Retry Interval and Retry Up Time parameters to reconnect multithreaded server components to the database if database connectivity has been lost.

Number of Sessions per SISNAPI Connection (alias SessPerSisnConn). Specifies how many sessions can be multiplexed (shared) through each SISNAPI connection, which helps to reduce the number of open network connections. SISNAPI connections are those between the Siebel Application Interface and the Application Object Manager. If Number of Sessions per SISNAPI Connection is -1, then all of the sessions are created through only one SISNAPI connection. The default value for this parameter is 20. However, while 20 is a good value to use for user sessions, it does not apply to incoming HTTP requests from other systems, for example, EAI HTTP Adapter access.

OM - Configuration File (alias ConfigFile). Specifies the configuration file used by the application associated to this component, such as siebel.cfg for Siebel Sales, uagent.cfg for Siebel Call Center, and so on. The setting of this parameter at the component level (such as an Application Object Manager, which is associated with an application) overrides the setting at the server level. See also *Overview of Parameters in Siebel Application Configuration Files*.

OM - Data Source (alias DataSource). Specifies the Application Object Manager data source.

OM - Named Data Source Name (alias NamedDataSource). Lists the named subsystems that the Application Object Manager preloads when it initializes. That is, all of the data sources that the Application Object Manager might use are listed in the value.

OM - Model Cache Maximum (alias ModelCacheMax). Specifies the size of the cache for model objects (also known as cached sessions) in object manager-based server components, such as Workflow Process Manager. Each model in the cache creates two database connections for the life of the model. (One connection is for insert, update, and delete operations. The other connection is for read-only operations.) The model cache improves performance for the object manager-based server components, and it is generally recommended not to disable this parameter. The default value is 10. A value of 0 disables this parameter, and the maximum setting for this parameter is 100. For more information about setting this parameter, see *Siebel Performance Tuning Guide* .

OM - Preload Links for batch components (alias PreloadLinksForBatchComp). Preloads all link definitions when an object manager-based server batch component starts. This advanced parameter is new as of Siebel CRM 20.7 Update. The default value is True. Preloading all link definition at component startup has the advantage of eliminating the same queries to fetch the link definitions and related definitions for business components and tables when the first delete is triggered on the component. Because all of these objects were preloaded at component startup, there is no need for them to be loaded on subsequent access. Setting this parameter to False can improve performance when starting object manager-based batch components, but degrades the performance for the first delete. Restarting the component is required when you change the value of this parameter.

OM - Resource Language Code (alias ResourceLanguage). Specifies the language code used for Application Object Manager resources. Each language has its own three-letter code identifier. For example, ENU identifies U.S. English. For more information, see *Siebel Global Deployment Guide* .

OM - Save Preferences (alias SavePreferences). Set the value for this parameter to False to prevent the Application Object Manager from creating or reading user preference files (file extension is .spf). When True (the default value),

the Application Object Manager creates a user preference file when a user logs in, if one does not already exist. In general, it is recommended to set SavePreferences to False for the EAI Object Manager. Please test and validate that your customizations do not require that SavePreferences be set to True.

OM - Timed Statistics (alias TimedStats). Enables the use of the statistics in the Administration - Server Management screen, Statistics view. Statistics include Average Connect Time (seconds), Average Response Time (milliseconds), and Average Think Time (seconds). Restart the Siebel Server for this parameter to take effect. The statistics are refreshed on completion of each task, so a specific user session must log out for the statistics for that session to be reflected in the Statistics screen. This is an advanced parameter.

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. The statistics data will be reset when the component is restarted.

Oracle Degree of Parallelism (alias OraDegreeOfParallelism). Allows setting the degree of parallelism for certain queries performed by Siebel Web Tools or Siebel Tools when used with an Oracle Database. See also the information about OraOptimizerIgnoreHints. For more information, see *Siebel Installation Guide* .

Oracle Optimizer Ignore Hints (alias OraOptimizerIgnoreHints). Specifies whether to ignore parallel hints in an Oracle Database. For parallel queries, do not change this parameter value from the default value, `False`, to `True`. See also the information about OraDegreeOfParallelism. For more information, see *Siebel Installation Guide* .

Password (alias Password). Specifies the database user password. This parameter specifies the password for the account referenced by the User Name parameter. Both the user name and password are prompted for during the initial configuration of the Siebel Server. The User Name and Password parameters are used to connect to the database for automatic startup of background mode components, for example. See also the description for the Auto Startup Mode parameter in *Siebel Server Parameters* and *Siebel Enterprise, Server, and Component Parameters*.

Process VM Usage Lower Limit (alias MemoryLimit). Sets the virtual memory usage threshold (in megabytes). A component process that reaches this threshold is recycled. The parameter to enable this feature is Memory Usage Based Multithread Shell Recycling. Use this parameter to remedy your application only if excessive memory usage created by memory leaks seems to be occurring.

Process VM Usage Upper Limit (alias MemoryLimitPercent). Specifies the percentage above the value of Process VM Usage Lower Limit at which a fast shutdown is triggered.

Retry Interval (alias RetryInterval). Defines the time interval before a series of retries are made for recovery. This parameter works with the Number of Retries and Retry Up Time parameters to reconnect multithreaded server components to the database if database connectivity has been lost.

Retry Up Time (alias RetryUpTime). Specifies the minimum up-time for a new set of retries for recovery. This parameter works with the Number of Retries and Retry Interval parameters to reconnect multithreaded server components to the database if database connectivity has been lost.

Session Keepalive Timeout (alias SessKeepAlive). Specifies the time that a task waits for a message from the client before timing out. When the time-out interval passes, the task ends the client session. The default value is 7200 seconds (two hours). This is a hidden parameter.

SISNAPI Connection Maximum Idle Time (alias ConnIdleTime). Configures the connection timeout between the Siebel Application Interface and the Siebel Server. Valid values are numeric, specifying the period of idle time (in seconds) after which the connection is disconnected by the component. See the following table for parameter setting descriptions. It is recommended that you set this parameter to a value slightly below the firewall connection timeout value. The default value is -1.

Parameter Value	Description
Less than 0	Disables this feature.
Any value between 0 and 30	Configures the minimum value for this parameter, 30 seconds.
Greater than 30	Configures that value in seconds.

Note: There is no direct relationship between this parameter and the session timeout setting for the Siebel Application Interface. The SISNAPI Connection Maximum Idle Time parameter controls the connection behavior while the session timeout setting controls the user session behavior. For more information, see *Siebel Installation Guide* .

SISNAPI - Log Traffic (alias LogTraffic). Specifies whether the Application Object Manager records a log of all of the Siebel Internet Session API (application programming interface) messages. SISNAPI is a session-based remote procedure call (RPC) designed to support high responsiveness between the Siebel Server and client applications.

Sleep Time (alias SleepTime). Specifies the time to sleep between iterations (in seconds). This parameter is used for the sleep time of component processes running in background mode when the Siebel Server is idle.

SQL Trace Flags (alias SQLFlags). The flags for tracing of SQL statements. If this parameter is set to 1, then every SQL statement issued by the component tasks is logged to the information log file for each task. If this parameter is set to 2, then each SQL statement is logged in addition to information about the number of parse, execute, and fetch calls, and timing information about each type of call.

Static Port Number (alias PortNumber). Specifies the network port number on which a component listens for client requests. If no value is specified, then a unique port number is generated dynamically for each component, starting at port number 49150 and higher, depending on the number of components configured and the occupied ports above this number.

The Siebel Connection Broker component (alias SCBroker), which provides intraserver load balancing, listens on the port specified by this parameter. For more information, see *About Siebel Connection Broker (SCBroker)*.

This parameter applies to interactive, batch, and background mode components, with the exception of all of the Application Object Manager components. The port number, whether static or dynamic, is hidden from the end user and is provided primarily so that administrators can fix the port numbers used by a component, for firewall configuration. If you are configuring this parameter, then select a port number either below 49150 or high enough to make sure that there are no conflicts with dynamically generated port numbers.

Trace Flags (alias TraceFlags). Specifies the flags for component-specific tracing information. This parameter is used to turn on various types of component-specific tracing. See the documentation for individual Siebel Server components for a description of how to set this parameter for each component.

Use IP Address (alias UseIPAddress). Specifies whether to construct the SISNAPI connection strings using the IP address instead of the hostname. The default value is True. Changing the value to False affects performance because the hostname must be looked up each time it connects.

Use Shared Log File (alias LogUseSharedFile). Specifies whether all of the tasks within a component process log to a shared file. When set to False, one log file for each task is generated.

User Name (alias Username). Specifies the database user name. This parameter specifies the user name of the database account that is used by the Siebel Server or Siebel Server components that are not started interactively or in batch mode by the Siebel Server Manager. The password for this database account must be specified by the Password

parameter. Both the user name and password are prompted for during the initial configuration of the Siebel Server. The User Name and Password parameters are used to connect to the database for automatic startup of background mode components and for other purposes. See also the description for the Auto Startup Mode parameter in *Siebel Server Parameters* and *Siebel Enterprise, Server, and Component Parameters*.

When you run component tasks from the Server Manager GUI, the value for the User Name parameter is used because the tasks are submitted as component jobs and started using the Server Request Broker server component, which uses the User Name parameter value. When you run component tasks from the Server Manager command-line interface, in the case of manually started batch or background mode components, the User Name parameter is that of the user who starts tasks on these components.

Correlating Database Sessions

The Object Manager Parameter Enable Database Session Correlation (`EnableDbSessCorrelation`) allows the Siebel Application to correlate a database connection with the database session created for the user.

When the database connection is created, this parameter logs the User Id that is associated with the connection allowing you to trace the connection's use by that user.

This is particularly useful when users report performance issues. You can look through the log to find the database session id associated with the user reporting the issue and find problematic SQL. You can also correlate user activity by querying `v$session` so that you can see what a session is doing, including its current SQL, wait events, and resource consumption. This is valuable for DBAs and performance analysts.

The following table describes `EnableDbSessCorrelation` Settings.

Parameter Name	Value	Description
`EnableDbSessCorrelation`	TRUE	The Database Session Id for a Connection will be logged.
`EnableDbSessCorrelation`	FALSE (Default)	No additional information will be logged.

The logging statements in the application log appear like this:

- Shared Connection Id: xxx
- Transaction Connection Id: yyyy
- "DataBase Connection Object was created at %x; DB User: '%s'"

11 Parameters in Siebel Application Configuration Files

Parameters in Siebel Application Configuration Files

This chapter includes information about some of the parameters in Siebel application configuration files. It includes the following topics:

- *Overview of Parameters in Siebel Application Configuration Files*
- *About Siebel Application Configuration Files*
- *Editing Siebel Application Configuration Files*
- *Configuration Parameters Index*
- *Description of Siebel Application Parameters*
- *Data Source Parameters*

Overview of Parameters in Siebel Application Configuration Files

This chapter includes information about the Siebel application configuration files and some of the parameters that these files might contain. Most of the parameters that are documented here are used by Siebel applications in the Siebel Mobile Web Client, or by Siebel Tools Client (using tools.cfg file).

In general, for Siebel Web Client applications connecting to an Application Object Manager component, the enterprise, server, component, or data source parameters stored in the Siebel Gateway registry, for example, are used instead of the corresponding parameters in the configuration file located on the Siebel Server. However, the Application Object Manager or Siebel Server might still use some configuration file parameters.

> **Note:** Not all of the parameters used by Siebel applications are described in this book. Many parameters are primarily described elsewhere on *Siebel Bookshelf* , while some parameters are not documented on *Siebel Bookshelf* .

Where you configure parameters for Siebel applications depends on your type of client deployment:

- For a Siebel Mobile Web Client deployment, you configure parameters in a configuration file (.cfg). Examples of configuration files include siebel.cfg, used by Siebel Sales, and uagent.cfg, used by Siebel Call Center. For more information, see *About Siebel Application Configuration Files* and *Editing Siebel Application Configuration Files*.
- For a Siebel Web Client deployment, you primarily configure parameters as component parameters for the Application Object Manager component or as named subsystem parameters. In either case, you use the Siebel Server Manager to configure the parameter. For more information, see *About Application Object Manager Parameters*.

The remaining topics in this chapter include the following:

- *About Siebel Application Configuration Files* lists some configuration files and their associated Siebel applications.
- *Editing Siebel Application Configuration Files* describes how you can edit a Siebel application configuration file.
- *Configuration Parameters Index* provides a partial list of configuration parameters and provides references to where they are documented in more detail.

  In many cases, when you set the equivalent parameter for an Application Object Manager component, for Siebel Web Clients, the alias of the parameter that appears in the Siebel Server Manager GUI is prefixed by CFG. The full parameter name also appears in the GUI.

  For example, for Siebel Mobile Web Clients, you set the parameter ApplicationSplashText in the configuration file to determine the text that appears on a splash screen when a user starts a Siebel application. For a Siebel Web Clients, you set the parameter Application Splashtext (alias CFGApplicationSplashText) on the relevant Application Object Manager component.
- *Description of Siebel Application Parameters* describes some of the parameters that are listed in *Configuration Parameters Index*.
- *Data Source Parameters* describes some of the data source parameters that are listed in *Configuration Parameters Index*.

Note: For Siebel Web Clients, many of these parameters can be set as named subsystem parameters for the relevant data source. In many cases, the parameter alias that appears in the Siebel Server Manager GUI is prefixed by DS. For example, the ConnectString parameter that you set in the `[ServerDataSrc]` section of an application configuration file appears as Data source Connect String (alias DSConnectString) for the Server Datasource named subsystem in the Siebel Server Manager GUI.

About Siebel Application Configuration Files

The following information lists several configuration files and their associated Siebel CRM applications. Your installation might contain additional configuration files besides those listed.

The configuration files are located in the `SIEBEL_CLIENT_ROOT\bin\LANGUAGE` directory on the Siebel Mobile Web Client. The name of the configuration file varies, depending on the application that you are using. Separate configuration files are provided for each supported language.

Siebel Application	Configuration File
Siebel Call Center	uagent.cfg
Siebel Marketing	market.cfg
Siebel Partner Manager	pmanager.cfg
Siebel Sales	siebel.cfg

Siebel Application	Configuration File
Siebel Service	sfs.cfg

The initial values for the parameters in the configuration files are either predefined by Siebel CRM or defined using the values that you specify during the initial configuration or at other times.

Some parameter values might include the notation $(param_name). These parameter values are substituted automatically, for example, during a client installation or local database initialization. For example, the value of the ConnectString parameter for a data source that has not yet been configured appears as follows:

```
ConnectString = $(ConnectString)
```

This parameter value is updated when you initialize the local database for a Siebel Mobile Web Client. For more information about initializing the local database, see *Siebel Remote and Replication Manager Administration Guide* . See also *Siebel Installation Guide* .

Editing Siebel Application Configuration Files

The Siebel application configuration files are plain-text files, and can be edited manually using a text editor. You can add parameters and their values or change values for existing parameters.

CAUTION: When you edit the configuration files, use a text editor that does not perform unwanted character substitutions or change the file's character encoding to formats other than UTF-8. For example, use Microsoft Notepad instead of Microsoft Word or WordPad.

There are many reasons why you might edit configuration files. For example, you might want to do so at some point after the installation and initial configuration in order to enable or disable certain functionality and features. You must make changes in all of the configuration files for the applications that you are using. For each new application that you create, you must copy and edit a configuration file to suit your needs.

Note: If a configuration parameter is not needed, then you can comment it out by inserting a semicolon at the start of the line.

To edit a Siebel application configuration file

1. Create a backup copy of the existing configuration file for which you want to edit parameter values, and save that file as a backup file.
2. Using any suitable text editor, such as Notepad, open the default version of the configuration file.
3. Edit parameter values, as necessary, to obtain the application behavior that you require.
4. Run a test using the configuration file that you edited.
    - If there is an error in your test, then correct the error and try again, or restore the configuration file from the backup file.
    - If no error occurs and the edit that you made has the result that you wanted, then use the modified configuration file.

Configuration Parameters Index

The following table contains an alphabetical listing of some of the configuration parameters from Siebel application configuration files such as siebel.cfg or uagent.cfg. This list shows the name of the section in which each parameter can be found and identifies where the parameter is documented (in tables in *Description of Siebel Application Parameters* and *Parameters for Individual Data Source Sections*). This list is not comprehensive.

Configuration Parameter Name	Section Name	Where Documented and Description
AccessDir	`[Siebel]`	*Description of Siebel Application Parameters*
AllowAnonUsers	`[InfraUIFramework]`	*Siebel Security Guide*
ApplicationName	`[Siebel]`	*Description of Siebel Application Parameters*
ApplicationPassword	Applicable security adapter sections, such as: `[LDAPSecAdpt]`	*Siebel Security Guide*
ApplicationSplashText	`[Siebel]`	*Description of Siebel Application Parameters*
ApplicationTitle	`[Siebel]`	*Description of Siebel Application Parameters*
ApplicationUser	Applicable security adapter sections	*Siebel Security Guide*
BaseDN	Applicable security adapter sections	*Siebel Security Guide*
CancelQueryTimeOut	`[InfraUIFramework]`	*Siebel Applications Administration Guide*
CaptionPrefix	`[Siebel]`	*Description of Siebel Application Parameters*

CaseInsensitive	Applicable data source sections, including:
	`[LOCAL_SE]`
	`[ServerDataSrc]`
	`[GatewayDataSrc]`
		Parameters for Individual Data Source Sections

Configuration Parameter Name	Section Name	Where Documented and Description
ClientRootDir	`[Siebel]`	*Description of Siebel Application Parameters*
CommConfigCache	`[Communication]`	*Siebel CTI Administration Guide*
CommConfigManager	`[Communication]`	*Siebel CTI Administration Guide*
CommConfigManagerName	`[Communication]`	*Siebel CTI Administration Guide*
CommEnable	`[Communication]`	*Siebel CTI Administration Guide*
CommLocalDriver	`[Communication]`	*Siebel CTI Administration Guide*
CommLogDebug	`[Communication]`	*Siebel CTI Administration Guide*
CommLogFile	`[Communication]`	*Siebel CTI Administration Guide*
CommMaxLogKB	`[Communication]`	*Siebel CTI Administration Guide*
CommMaxMsgQ	`[Communication]`	*Siebel CTI Administration Guide*
CommReleaseLogHandle	`[Communication]`	*Siebel CTI Administration Guide*
CommReqTimeout	`[Communication]`	*Siebel CTI Administration Guide*
ComponentName	`[Siebel]`	*Description of Siebel Application Parameters*

ConnectString	Applicable data source sections, including:
	`[LOCAL_SE]`
	`[ServerDataSrc]`
	`[GatewayDataSrc]`
		Parameters for Individual Data Source Sections (data sources)

ContactLogin	Applicable data source sections	*Parameters for Individual Data Source Sections*
CredentialsAttributeType	Applicable security adapter sections	*Siebel Security Guide*
DataSource	`[Siebel]`	*Description of Siebel Application Parameters*

Configuration Parameter Name	Section Name	Where Documented and Description
DebugLevel	`[EMail]`	*Siebel Email Administration Guide*
DefaultMailClient	`[EMail]`	*Siebel Email Administration Guide*
DisableExecuteRetry	`[ServerDataSrc]`	*Parameters for Individual Data Source Sections*
DisableReverseProxy	`[InfraUIFramework]`	*Siebel Security Guide*

DLL	Applicable data source sections, including:
	`[LOCAL_SE]`
	`[ServerDataSrc]`
	`[GatewayDataSrc]`
		Parameters for Individual Data Source Sections

Docked	Applicable data source sections, including:
	`[LOCAL_SE]`
	`[ServerDataSrc]`
		Parameters for Individual Data Source Sections

DockConnString	`[LOCAL_SE]`	*Parameters for Individual Data Source Sections*
DockRecvTxnsPerCommit	Applicable data source sections	*Parameters for Individual Data Source Sections*
DockRepositoryName	`[Siebel]`	*Description of Siebel Application Parameters*
DockTxnsPerCommit	`[LOCAL_SE]`	*Parameters for Individual Data Source Sections*
EditFieldCaption	`[InfraUIFramework]`	*Configuring Siebel Business Applications*
EditFieldType	`[InfraUIFramework]`	*Configuring Siebel Business Applications*
Enable	`[DataCleansing]` `[DeDuplication]`	*Siebel Data Quality Administration Guide*
EnableFQDN	`[Siebel]`	*Siebel Installation Guide*

Configuration Parameter Name	Section Name	Where Documented and Description
EnableInlineForList	`[InfraUIFramework]`	If the applet mode property is set to Edit List, then the functions Edit, New, and Query work in-line, that is, in the list applet, when EnableInlineForList is set to True. The default value is True. Set to False to turn off this behavior.
EnablePersonalization	`[Siebel]`	*Description of Siebel Application Parameters*
EnableScripting	`[Siebel]`	*Description of Siebel Application Parameters*

EnterpriseServer	Applicable data source sections, including:
	`[LOCAL_SE]`
	`[ServerDataSrc]`
		Parameters for Individual Data Source Sections

eProdCfgNumbOfCachedCatalogs	`[InfraObjMgr]`	*Siebel Performance Tuning Guide*
eProdCfgNumbofCachedWorkers	`[InfraObjMgr]`	*Siebel Performance Tuning Guide*
eProdCfgNumOfCachedAttrs	`[InfraObjMgr]`	*Siebel Performance Tuning Guide*
eProdCfgNumOfCachedClasses	`[InfraObjMgr]`	*Siebel Performance Tuning Guide*
eProdCfgSnapshotFlg	`[InfraObjMgr]`	*Siebel Performance Tuning Guide*
FileSystem	Applicable data source sections, including: `[ServerDataSrc]`	*Parameters for Individual Data Source Sections*
FQDN	`[Siebel]`	*Siebel Installation Guide*
GatewayAddress	Applicable data source sections, including: `[ServerDataSrc]`	*Parameters for Individual Data Source Sections*
GatewayDataSrc	`[DataSources]`	*Parameters for Data Sources Section*
Hidden	Applicable data source sections, including: `[GatewayDataSrc]`	*Parameters for Individual Data Source Sections*

Configuration Parameter Name	Section Name	Where Documented and Description

InsensitivityFactor	Applicable data source sections, including:
	`[LOCAL_SE]`
	`[ServerDataSrc]`
	`[GatewayDataSrc]`
		Parameters for Individual Data Source Sections

InsUpdAllCols	Applicable data source sections	*Parameters for Individual Data Source Sections*
IntegratedSecurity	Applicable data source sections	*Parameters for Individual Data Source Sections*
Local	`[DataSources]`	*Parameters for Data Sources Section*
LocalDbODBCDataSource	`[Siebel]`	*Description of Siebel Application Parameters*
LoginView	`[InfraUIFramework]`	*Siebel Security Guide*

MaxCachedCursors	Applicable data source sections, including:
	`[LOCAL_SE]`
	`[ServerDataSrc]`
		Parameters for Individual Data Source Sections

MaxCachedDataSets	Applicable data source sections, including:
	`[LOCAL_SE]`
	`[ServerDataSrc]`
		Parameters for Individual Data Source Sections

MaxConnections	Applicable data source sections	*Parameters for Individual Data Source Sections*
MaxCursorSize	Applicable data source sections, including: `[ServerDataSrc]`	*Parameters for Individual Data Source Sections*
MaxDepthOfTheQuery	`[InfraObjMgr]`	*Description of Siebel Application Parameters*
MessageBarUpdateInterval	`[Siebel]`	*Siebel Applications Administration Guide*

Configuration Parameter Name	Section Name	Where Documented and Description
MultiCurrency	`[Siebel]`	*Description of Siebel Application Parameters*
NonSQL	Applicable data source sections	*Parameters for Individual Data Source Sections*
NumberOfListRows	`[InfraUIFramework]`	*Siebel Object Types Reference*
PasswordAttributeType	Applicable security adapter sections	*Siebel Security Guide*
PersonalizationLog	`[Siebel]`	*Description of Siebel Application Parameters*
Port	Applicable security adapter sections	*Siebel Security Guide*
PrefetchSize	Applicable data source sections, including: `[ServerDataSrc]`	*Parameters for Individual Data Source Sections*
PrimaryEnterprise	Applicable data source sections, including: `[GatewayDataSrc]`	*Parameters for Individual Data Source Sections*
RemoteSearchServer	`[Siebel]`	*Description of Siebel Application Parameters*
RemoteSearchServerPath	`[Siebel]`	*Description of Siebel Application Parameters*
ReportDataDir	`[XMLPReports]`	*Siebel Reports Guide*
ReportOutputDir	`[XMLPReports]`	*Siebel Reports Guide*
ReportsDir	`[Siebel]`	*Description of Siebel Application Parameters*
ReportsODBCDataSource	`[Siebel]`	*Description of Siebel Application Parameters*
RepositoryFile	`[Siebel]`	*Description of Siebel Application Parameters*
RequestServerName	`[InfraObjMgr]`	This parameter is the name of the Siebel Server that services requests from the Siebel client.

Configuration Parameter Name	Section Name	Where Documented and Description
RequiredIndicator	`[InfraUIFramework]`	*Siebel Developer's Reference*

ReverseFillThreshold	Applicable data source sections, including:
	`[LOCAL_SE]`
	`[ServerDataSrc]`
		Parameters for Individual Data Source Sections

RolesAttributeType	Applicable security adapter sections	*Siebel Security Guide*
Sample	`[DataSources]`	*Parameters for Data Sources Section*
ScriptingDLL	`[Siebel]`	*Description of Siebel Application Parameters*
SearchDefName	`[Siebel]`	*Description of Siebel Application Parameters*
SearchEngine	`[Siebel]`	*Description of Siebel Application Parameters*
SearchInstallDir	`[Siebel]`	*Description of Siebel Application Parameters*
SecAdptDllName	Applicable security adapter sections	*Siebel Security Guide*
SecAdptMode	`[InfraSecMgr]`	*Siebel Security Guide*
SecAdptName	`[InfraSecMgr]`	*Siebel Security Guide*
ServerDataSrc	`[DataSources]`	*Parameters for Data Sources Section*
ServerName	Applicable security adapter sections	*Siebel Security Guide*
SharedCredentialsDN	Applicable security adapter sections	*Siebel Security Guide*
SharedModeUsersDir	`[Siebel]`	*Description of Siebel Application Parameters*
ShowMessageBar	`[Siebel]`	*Siebel Applications Administration Guide*
ShowWriteRecord	`[InfraUIFramework]`	*Siebel Partner Relationship Management Administration Guide*

Configuration Parameter Name	Section Name	Where Documented and Description
SiebelExtMailClientAttDir	`[EMail]`	*Siebel Email Administration Guide*
SiebelUsernameAttributeType	Applicable security adapter sections	*Siebel Security Guide*
SingleSignOn	Applicable security adapter sections	*Siebel Security Guide*

SqlStyle	Applicable data source sections, including:
	`[LOCAL_SE]`
	`[ServerDataSrc]`
		Parameters for Individual Data Source Sections

SslDatabase	Applicable security adapter sections	*Siebel Security Guide*
SystemSWFName	`[InfraUIFramework]`	*Configuring Siebel Business Applications*
SystemSWSName	`[InfraUIFramework]`	*Configuring Siebel Business Applications*

TableOwner	Applicable data source sections, including:
	`[LOCAL_SE]`
	`[ServerDataSrc]`
		Parameters for Individual Data Source Sections

TempDir	`[Siebel]`	*Description of Siebel Application Parameters*
TreeNode*X* (several parameters that are similarly named, for example, TreeNodeCollapseCaption)	`[InfraUIFramework]`	*Configuring Siebel Business Applications*
TrustToken	Applicable security adapter sections	*Siebel Security Guide*
Type	`[DataCleansing]` `[DeDuplication]`	*Siebel Data Quality Administration Guide*
UpperCaseLogin	Applicable data source sections	*Parameters for Individual Data Source Sections*
UseDictionaryinTransactionLogging	Applicable data source sections	*Parameters for Individual Data Source Sections*
UseAdapterUsername	Applicable security adapter sections	*Siebel Security Guide*

Configuration Parameter Name	Section Name	Where Documented and Description
UseRemoteConfig	Applicable security adapter sections	*Siebel Security Guide*
UsernameAttributeType	Applicable security adapter sections	*Siebel Security Guide*
UserSWFName	`[InfraUIFramework]`	*Configuring Siebel Business Applications*
UserSWSName	`[InfraUIFramework]`	*Configuring Siebel Business Applications*
Version	`[Siebel]`	*Description of Siebel Application Parameters*
WebTemplatesVersion	`[InfraUIFramework]`	*Siebel Performance Tuning Guide*
XdoDir	`[XMLPReports]`	*Siebel Reports Guide*

Description of Siebel Application Parameters

The following table describes several Siebel application parameters. These parameters appear in the [Siebel] section of the configuration file and apply to Siebel Mobile Web Client or Siebel Developer Web Client.

For the Siebel Web Client, the parameters are defined as server parameters on the Application Object Manager, rather than defined in a configuration file. Some parameters also apply to Siebel Tools and might be defined in the tools.cfg file.

Name	Description
AccessDir	Specifies the directory where Microsoft Access is installed.
ApplicationName	Name of the application object in the repository to use, such as Siebel Sales or Siebel Service. ApplicationName determines which splash screen appears and which set of menus is enabled.
ApplicationSplashText	Text that appears on a splash screen when starting up a Siebel application. The default varies by the application. If you are starting Siebel Sales, for example, then ApplicationSplashText is set to Siebel Sales by default.
ApplicationTitle	Changing the text in the ApplicationTitle parameter in the configuration file changes the Application Title value in the multi-value group applets and the Title that appears on the application title bar. As necessary, make this change in the appropriate configuration file for the intended application. For example, for Siebel Call Center, this parameter is set, in the file uagent.cfg, to *Siebel Call Center*.
CaptionPrefix	Allows customizing of the title in the corner of the Siebel client application. Reads Siebel by default.

Name	Description
ClientRootDir	Specifies the directory where the Siebel client software is installed.
ComponentName	Specifies the Siebel Anywhere configuration that is used during version check. Navigate to the Administration - Siebel Anywhere screen, then the Configurations view, to see the configurations. Change the setting for this parameter if you want this configuration file to be version-checked by a specific Siebel Anywhere configuration. For example, if you want to check the version of Siebel CRM applications used at one facility with the version used at another facility, then create two Siebel Anywhere configurations appropriately named. Then, in the configuration file for each set of users, enter a different value for the ComponentName parameter. It is recommended that you use only alphanumeric characters plus dashes and normal parentheses for the ComponentName parameter. If you want to use the Priority upgrade feature, then the maximum character length for the Upgrade Component name is 40.
DataSource	Name of the default data source that appears in the Connect To drop-down list in the Siebel login screen. Must correspond to an existing data source defined in the configuration file. By default, this parameter is set to Local.
DockRepositoryName	The value matches the Siebel repository used for generating the Siebel runtime repository.
EnablePersonalization	Must be set to True to activate the personalization (content targeting) functionality. Siebel workflow processes will not execute properly until personalization events are reloaded.
EnableScripting	True or False. Enables use of Siebel Visual Basic or Siebel eScript. Setting this parameter to False also disables browser scripts in addition to server scripts.
LocalDbODBCDataSource	Name of the ODBC data source that is set up to access the local database. It is used by a variety of features in the Siebel CRM applications and in Siebel Tools. Do not modify it, because it is already correctly configured for the Siebel client or Siebel Tools.
MaxDepthOfTheQuery	Allows you to increase the maximum parse depth of a query with a large search specification, including, for example, a large number of OR or AND clauses. You can set this parameter in the application configuration file for the Siebel Mobile Web Client or Developer Web Client, or set it in the tools.cfg file for Siebel Tools. Set it in the [InfraObjMgr] section. Alternatively, you can set this parameter for the Siebel Enterprise. The full name of the server parameter is Maximum depth of the query. For more information about this parameter, see *Siebel Enterprise, Server, and Component Parameters* (see *Generic Parameters*).
MultiCurrency	True or False. Enables multicurrency support.
PersonalizationLog	Add the following to the configuration file to view a log of all personalization activity: `PersonalizationLog = "C:\personalization.txt"`

Name	Description
	where C: is the drive where you want to store the log. The log can help you debug your rules, events, and actions.
RemoteSearchServer	True or False. True indicates that searches are performed on a remote computer. False indicates that searches are performed on a local computer.
RemoteSearchServerPath	Indicates the name of the remote computer that performs searches.
ReportsDir	Directory where reports are installed. It is typically left blank.
ReportsODBCDataSource	Name of ODBC data source used to connect to external modules.
RepositoryFile	Name of the Siebel runtime repository to use.
ScriptingDLL	Name of the shared library that implements Siebel Visual Basic or Siebel eScript. If the Siebel Server runs on a UNIX server computer, and you plan to use Siebel eScript, then set the value of ScriptingDLL to sscfjs.so.
SearchDefName	Search definition from Siebel Tools to be used for searching. For more information about search, see *Siebel Search Administration Guide* .
SearchEngine	Defines the search engine to use for search. You set this parameter to the value that identifies the search engine. If your search engine is Oracle Secure Enterprise Search, then you set it to **`SES`**. For more information about search engine administration, see *Siebel Search Administration Guide* .
SearchInstallDir	Set this parameter to the directory where your search engine is installed. For more information about search engine administration, see *Siebel Search Administration Guide* .
SharedModeUsersDir	Directory where user preference files are stored. It is typically left blank. Otherwise, it must be set to a directory under a network shared drive, but it cannot be set to the shared drive itself. For example, if **`\\yourserver\common`** is the network shared drive, then you cannot set SharedModeUsersDir to \\yourserver\common. Instead, set SharedModeUsersDir to a directory under **`\common`**.
TempDir	Directory where temporary files are created.
Version	Represents the version of a file. This parameter is for internal use only and is automatically maintained by Siebel Anywhere. When you create an upgrade kit for the configuration file, Siebel Anywhere increments this version string appropriately, based on the version information from the Upgrade Components view. **Note:** When you perform a Siebel Anywhere upgrade, you must manually upgrade files from the upgraded Siebel client directory to the Siebel Server directory.

Data Source Parameters

This topic describes parameters that pertain to the [DataSources] section of the configuration file and to the individual data source sections that follow.

These parameters apply only to Siebel Mobile Web Client. For the Siebel Web Client. The parameters are defined as server parameters on the Application Object Manager.

This topic contains the following information:

- *Parameters for Data Sources Section*
- *Parameters for Individual Data Source Sections*

Parameters for DataSources Section

The following table lists data sources in the Siebel configuration file, which appear in the [DataSources] section. Each data source also has its own section specifying parameters that apply to this data source.

Name	Function
Local	Defines parameters for connecting to the local database.
Sample	Defines parameters for connecting to the sample database.
ServerDataSrc	Defines parameters for connecting to the Siebel database.
GatewayDataSrc	Defines Siebel Gateway parameters.

Note: If you want to prevent a data source from being displayed as a choice in the Connect To: portion of the Siebel login screen, then add two slash characters (//) in front of the data source in the [DataSources] section of the configuration file. For example: `//Sample = Sample`.

Parameters for Individual Data Source Sections

The following table lists parameters that specify properties associated with the different data sources listed under [DataSources]. Each data source section defines the properties of the particular data source.

Name	Comment
CaseInsensitive	True or False. If it is True, then the client is notified to work with the database in case-insensitive mode. See also the description of the InsensitivityFactor parameter for data sources.

Name	Comment
	Note: Queries against fields of type DTYPE_ID are always case-sensitive, even if the CaseInsensitive parameter is set to True. For more information, see *Siebel Applications Administration Guide* .
ConnectString	Database-dependent string that defines how to connect to the database (also referred to as the *connection string*). The ConnectString parameter is also used to specify the Siebel Gateway computer in the GatewayDataSrc section. In the Siebel Mobile Web Client's configuration file, you must specify the Siebel Gateway's hostname, preferably in a fully qualified form like *node.domain*.**xxx**. Failure to specify this parameter correctly results in the server administration screens being inaccessible. For more information about using connection strings for different server databases, see *Siebel Installation Guide* .
ContactLogin	True or False. If it is True, then it indicates that the corresponding data source uses contact login, rather than employee login. Because a contact user is generally not associated one-to-one with a database account, you must use a security adapter to support contact users. If it is False, then the data source is using employee login, rather than contact login.
DisableExecuteRetry	True or False. When set to True, this parameter prevents the client from resending a query to the database layer if an error occurs such as a network error or session kill.
DLL	Name of the DLL file to use for the database connector code. The names differ depending upon whether you are using Oracle Database, Microsoft SQL Server, IBM DB2, and so on.
Docked	Determines which database connection the application login screen for a Mobile Web Client defaults to. For example, if Docked is True in the [ServerDataSrc] section of the configuration file and Docked is False in the [LOCAL_SE] section of the configuration file, then the application login screen defaults to the server database rather than the local database. The values used in this example are the default values.
DockConnString	Name of the docking server (Siebel Remote Server). It is the computer name of the Siebel Server against which the Mobile Web Client synchronizes. For information about how to format the values for this parameter, see *Siebel Remote and Replication Manager Administration Guide* .
DockRecvTxnsPerCommit	Number of transactions received by the Mobile Web Client before a commit is issued to the database. The default value for this parameter is 10. Change the setting to: • A higher value if you have a fast network connection, such as a LAN. Increasing the value can provide better performance when synchronizing the Mobile Web Client with the server. • A lower value if you have a lower-bandwidth network connection, such as a modem.
DockTxnsPerCommit	Number of transactions processed before a commit is issued to the database.
EnterpriseServer	Name of the Siebel Enterprise Server.
FileSystem	Specifies how the Mobile or Siebel Developer Web Client (for administrative purposes) accesses the Siebel File System. The value of this parameter cannot exceed 2048 characters. Generally, FileSystem and other parameters identified in this table are set during the Siebel client installation.

Name	Comment
	Mobile Web Client. The following scenario for setting the FileSystem parameter applies to the Siebel Mobile Web Client. Install the Siebel File System locally on a Mobile Web Client, so that it is accessible when the client is not connected to the network and can be synchronized using Siebel Remote: Set the following parameter, where FS_location is a UNC location or a drive-letter path to the location on the client computer where the local Siebel File System was installed: FileSystem = FS_location **Developer Web Client.** The following scenarios for setting the FileSystem parameter apply to the Siebel Developer Web Client when used for administrative purposes. Make sure that users on the Siebel Developer Web Client have physical access privileges for the Siebel File System directories: • If the Developer Web Client installation uses File System Manager (alias FSMSrvr), then set the following parameters: FileSystem = *FSM* GatewayAddress = Siebel_Gateway_hostname EnterpriseServer = Siebel_Enterprise_Server_name • If the Developer Web Client installation does *not* use FSMSrvr, then set the following parameter, where FS_location is a UNC location or a drive-letter path to the location on a network computer where the Siebel File System was installed: FileSystem = FS_location If your networked Siebel File System resides on a UNIX server, then you require a cross-platform NFS file system mounting tool to connect from Siebel Developer Web Clients running on Windows computers.
GatewayAddress	Host name or IP address of the Siebel Gateway.
Hidden	True or False. Determines if the data source shows up in the login screen's picklist of data sources.
InsensitivityFactor	Set to a positive integer value (default is 2). Applies only when the CaseInsensitive parameter is True for the data source. The value controls the number of characters in each string that are treated as case-insensitive in a query. Not all database vendors support case-insensitivity efficiently, so this feature provides an approximate solution. See also the description of the CaseInsensitive parameter for data sources. The following is an example SQL WHERE clause generated when searching for an opportunity named New, when InsensitivityFactor is set to 2. WHERE ((S_OPTY.NAME LIKE 'ne%' OR S_OPTY.NAME LIKE 'Ne%'OR S_OPTY.NAME LIKE 'nE%'OR S_OPTY.NAME LIKE 'NE%') AND UPPER(S_OPTY.NAME)=UPPER('New'))

Name	Comment
	This example shows that all of the permutations of the first two letters of the string *New* are checked. With a higher factor, the number of permutations grows exponentially, and performance suffers. **Note**: Do not set this parameter to a value higher than 13.
InsUpdAllCols	True or False. Ordinarily, when the Siebel application generates INSERT or UPDATE statements to send to the database, the actual statement contains only the columns where data is present or has changed. When there are situations where you generate many statements on a particular table, the differences in the values being updated might prevent you from using an array interface supported by the DBMS. When this feature is set to True, all of the columns are present in all INSERT and UPDATE statements. Where two statements are issued against the same table in the same business component as part of a batch operation, this setting automatically enables the statements to use any existing array feature of the DBMS.
IntegratedSecurity	True or False. When True, the Siebel client is prevented from prompting the user for a user name and password when the user logs in. Facilities provided in your existing data server infrastructure determine whether the user is allowed to log into the database. This parameter is set for your server data source. However, it is supported for Oracle Database and Microsoft SQL Server databases only. The default value is False. For additional information, refer to your third-party documentation. For Oracle Database, refer to the OPS$ and REMOTE_OS_AUTHENT features. For Microsoft SQL Server, refer to Integrated Security.
MaxCachedCursors	Specifies the maximum number of SQL cursors that can be cached in memory for a database connection. The default is 16. Caching SQL cursors can improve response time and CPU usage because an SQL cursor does not have to be prepared each time it is executed. If memory usage is not a concern, then you might consider increasing the value of this parameter.
MaxCachedDataSets	Specifies the maximum number of data sets that can be cached in memory for a database connection. The default is 16. A data set is the set of records that has been retrieved by the execution of a business component. Data-set caching applies only to those business components for which the Cache Data property has been set in Oracle's Siebel Tools. Caching data sets for frequently visited business components can improve response time and CPU usage. If memory usage is not a concern, then you might consider increasing the value of this parameter.
MaxConnections	Number of connections that can be made to the data source database server.
MaxCursorSize	Sets the total number of rows that can be returned in a result set. MaxCursorSize is intended for use *only* with IBM DB2 for OS/390 and z/OS, and must be set as described in *Implementing Siebel Business Applications on DB2 for z/OS* . If you are using another database, then do not set this parameter to any value other than the default value (-1), or database behavior is adversely affected. MaxCursorSize and PrefetchSize are used together, and they must be set to the same value. See also the description for the PrefetchSize parameter.
NonSQL	True or False. Setting that indicates that the data source does not use an SQL DBMS to retrieve its data. Use this parameter only in conjunction with a specialized business component that your company builds internally. Never use this parameter arbitrarily.

Name	Comment
PrefetchSize	Sets the number of rows that the Siebel application reads initially as part of a query execution. PrefetchSize is intended for use *only* with IBM DB2 for OS/390 and z/OS, and must be set as described in *Implementing Siebel Business Applications on DB2 for z/OS* . If you are using another database, then do not set this parameter to any value other than the default value (-1), or database behavior is adversely affected. MaxCursorSize and PrefetchSize are used together, and they must be set to the same value. See also the description for the MaxCursorSize parameter.
PrimaryEnterprise	The name of the Enterprise Server that you want to administer from the client computer. Set this parameter to view or change information in the server administration views.
ReverseFillThreshold	When the current query contains many rows, it might be very inefficient for the user to click End try to display and read them all sequentially. For this reason, the customer might configure a threshold value to invert the current sort, re-execute the query, and fill the data buffers from the end. These internal operations are hidden from the user.
SqlStyle	Indicates what kind of SQL to send to the database that you are using. When generating SQL to send to a DBMS, the application constructs the SQL statement to suit the particular DBMS. The value of this parameter is automatically set by the Siebel client installer or by initial server configuration, according to database information that you specified. The local database, based on Oracle Database SE2, uses OracleCBO (the same as Oracle Database Enterprise). IBM DB2 and Microsoft SQL Server use the style applicable to the particular DBMS.
TableOwner	In a database, tables are identified by both their owner and their name. When queries that reference tables are issued, the table owner must be included in those references (for example, SIEBEL.S_EVT_ACT, where SIEBEL is the table owner). Siebel Server infrastructure and system management components, such as SRBroker and SRProc, read the value of this parameter. For Application Object Manager server components, you specify the table owner by setting a value for the Datasource Table Owner (alias DSTableOwner) named subsystem parameter.
UpperCaseLogin	The default is False. If it is set to True, then the user ID, when a user logs in, is converted to uppercase before it is sent to the database for authentication. This value is applicable only if the database is used for authentication. The value of the parameter is ignored when SecurityAdapter is set to a nonempty value. Use this parameter if you want to enforce a policy of having all of the database accounts in uppercase on a case-sensitive database, but you do not want users to worry about case when they type in their user names. **Note**: The value of UpperCaseLogin does not affect the password.
UseDictionaryinTransaction Logging	To avoid the diccache.dat creation while using the Developer Web Client connected to the database server, add this parameter to the [ServerDataSrc] section in the configuration file. `UseDictionaryInTransactionLogging = False` **Note:** This parameter is not supported if you plan to use Siebel Remote and synchronize between client and server. If you plan to perform only testing and administration tasks connected against the database server, then you will not be affected. For Siebel Remote testing, use the Siebel Web Client and the Siebel Mobile Web Client.

12 Siebel CRM Event Publication and Subscription

Siebel CRM Event Publication and Subscription

This chapter includes information about some of the parameters in Siebel application configuration files. It includes the following topics:

- *Overview of Siebel CRM Event Publication and Subscription*
- *Configuring Synchronous Publishing*
- *Information Flow in Asynchronous Publishing*
- *Summary of Event Publication and Subscription Setup Steps*
- *Configuration Files for Event Publication and Subscription*
- *OCI Streaming with Apache Kafka - Configurations for Integrating Siebel Event Publication and Subscription*
- *Unit Testing Event Publication and Subscription*
- *Kafka Consumer Application Scalability Guidelines*
- *Avro Serialization in Siebel CRM Kafka Integration*
- *Recommended Operational Practices*
- *Troubleshooting and Debugging*

Overview of Siebel CRM Event Publication and Subscription

An event is a change of state. For example, 'Account Status' changing from 'Unverified' to 'Verified', new customer creation; existing customer updates; new items in cart or, for that matter, any other change that is important for business.

The Siebel CRM Event Publication and Subscription feature enables you to send event information from Siebel to Kafka and receive event information into Siebel CRM from Kafka. It supports high throughput message exchange between Siebel and Kafka in both synchronous and asynchronous manners.

As a direct benefit, it allows for the decoupling of direct interaction between Siebel and integrated applications, which means, event producers applications (for example Siebel CRM) and consumer applications (for example, payment processing system) can work independently of each other and focus on their domains of operation.

Dedicated Application Interfaces (AIs) can be used to communicate between Siebel Object Managers and Kafka server using Consumer and Producer Java APIs of Kafka Client. These AIs are referred as sidecar AIs throughout this document.

These Sidecar AIs are standard Siebel CRM Application Interfaces that have additional configuration for Siebel CRM Kafka integration for the Siebel CRM Event Publication and Subscription feature. Siebel CRM Event Publication and

Subscription functionality therefore works only for those components that are accessible from Application Interface. For example: session-based or interactive components and not of type batch.

The word Sidecar in Sidecar AI refers to the usage of sidecar design pattern, in that the functionalities included in those AIs that are used as sidecars enhance the Siebel core platform functionalities and aid in high throughout event exchange between Siebel CRM and Kafka with easy upscaling as may be required. Therefore, the reference of sidecar pods in a Kubernetes deployment (for example obtained by using Siebel Cloud Manager on OCI OKE) do not apply here and these sidecar AIs are to be deployed as separate containers or pods in the cloud, like traditional AIs in a Kubernetes deployment of Siebel CRM. The installation and configuration process for a sidecar AI therefore is same as that of traditional Siebel AIs, along with additional configurations relevant for Siebel Kafka integration.

Oracle provides Kafka Client as part of Siebel installation. Kafka server (also referred to as broker in this document) needs to be installed and managed by the user. Kafka Client and Server(broker) versions can be independently upgraded (Refer to *KIP-97*). It is recommended to independently upgrade Kafka servers immediately upon discovery of any critical vulnerability.

The Siebel CRM Kafka integration is implemented over the Java Client APIs of the Kafka distribution from Apache foundation. The producer and consumer Java APIs are implemented within Siebel CRM AIs. If any other Kafka distribution has support for the Java producer and consumer APIs that are from the main Kafka project, this integration may work with Kafka brokers/servers of such distributions. However, since Oracle will not know the implementation details, support from Oracle will be limited in such cases.

The Siebel CRM Kafka integration also supports Avro Serialization. This feature helps you to optimize messages exchanged between the Kafka broker and Kafka client. For more information, see *Avro Serialization in Siebel CRM Kafka Integration*.

Business Service and Class

An important component of Kafka integration in Siebel CRM Event Publication and Subscription is a business service **Event Handler**. It is based on C++ Class *CSSEAIEventHandler.* One of its methods, **SendEvent**, does the job of posting Siebel events to Kafka. For SendEvent method, input is Siebel Property Set data with target Kafka Topic name at the beginning. It posts JSON to Kafka topic.

SendEvent Method's Input Sample (a simplified Siebel Property Set equivalent representation)

```
{
 p["topic"] = "CreateContactEvent";
 p["partitions"] = "0,1,3";
 p["BusinessObjName"] = "Contact";
 {
 p["BusinessCompName"] = "Contact";
 {
 p["First Name"] = "John";
 p["Last Name"] = "Doe";
 p["Contact Id"] = "88-37WGM9";
 }
 }
 {
 p["BusinessCompName"] = "Account";
 {
 {
 p["City"] = "Menlo Park";
 p["CSN"] = "10-103";
 p["Location"] = "HQ-Corporate";
 p["Name"] = "A. K. Parker Inc.";
 }
 }
 }
}
```

SendEvent Method's Output Sample JSON

```
{
 "Contact": {
 "Fields": {
 "Contact Id": "88-37WGM9",
 "First Name": "John",
 "Last Name": "Doe"
 },
 "BusinessCompName": "Contact"
 },
 "Account": {
 "Field List": {
 "Fields": {
 "Name": "A. K. Parker Inc.",
 "Location": "HQ-Corporate",
 "CSN": "10-103",
 "City": "Menlo Park"
 }
 },
 "BusinessCompName": "Account"
 },
 "BusinessObjName": "Contact"
}
```

The Event Publication and Subscription feature requires the use of business services based on classes derived from *CSSEAIEventHandler*.

From the Siebel 24.12 monthly update release, Siebel CRM Kafka integration through Siebel CRM Event Publication and Subscription supports an optional new payload structure to post messages to Kafka using the Run-Time Events (RTE) mechanism.

You can control the payload structure (old/new) to post messages to Apache Kafka through RTE mechanism using an optional environment variable `SIEBEL_EVENT_PUBSUB_PAYLOAD_STRUCT` . This variable can take one of the following values:

- **1** - This is the default value and specifies that the old payload structure will be used to post messages to Kafka using the RTE mechanism. If the value of `SIEBEL_EVENT_PUBSUB_PAYLOAD_STRUCT` is set to an invalid value or left empty, it will default to 1.

  Sample use case and default (old) payload:

  - Use case
    - Primary Business Component: Contact
    - Child Business Component Data: One Account
  - Payload:

```
{
 "Contact": {
 "Fields": {
 "Contact Id": "101",
 "First Name": "PayLoad",
 "Last Name": "EPSSiebel",
 },
 "BusinessCompName": "Contact"
 },
 "Account": {
 "Field List": {
 "Fields": {
 "Name": "A. K. Parker Inc.",
 "Location": "HQ-Corporate",
 "CSN": "10-103",
```

```
 "City": "Menlo Park"
 }
 },
 "BusinessCompName": "Account"
 },
 "BusinessObjName": "Contact"
}
```

- **2** - Specifies that the new payload structure will be used to post messages to Kafka using RTE. This new message structure is optional and in this format Siebel CRM will:
  - Not send the business component name (`BusinessCompName`) and business object name (`BusinessObjName`) in the payload structure.
  - Post primary business component as JSON object.
  - Determine how a single data record of a child business component is posted using the value of the `SingleRecordAsJSONObject` property.

  Sample use case and (new optional) payload:

  - Use case
    - Primary Business Component: Contact
    - Child Business Component Data: One Account
  - Payload

```
{
 "Contact": {
 "Contact Id": "101",
 "First Name": "PayLoad",
 "Last Name": "EPSSiebel",
 "Account": [
 {
 "Name": "A. K. Parker Inc.",
 "Location": "HQ-Corporate",
 "CSN": "10-103",
 "City": "Menlo Park"
 }
 ]
 }
}
```

Note: The old JSON payload structure, is the default and will continue to work but support for the same will be dropped in the future releases. Hence, it is recommended that, you transition to the new payload structure at the earliest.

Siebel CRM Event Publication

You can publish messages from Siebel CRM in two ways:

1. **Synchronously**- Also known as the 1-step publish, where whole payload is sent to Kafka in one go. For more information see, *Configuring Synchronous Publishing*
2. **Asynchronously**- Also known as the 2-step publish. Here, the main component where an event is generated sends only the record identifier, while another (cloned) component handles preparation of full event messages, which frees the main component to serve critical business tasks without introducing significant latency. This also helps in scaling up for higher volume message exchange. The first of 2-step publish utilizes synchronous

publish, and the second step utilizes subscription mechanism. For more information see, *Information Flow in Asynchronous Publishing* .

Siebel CRM Event Subscription

Siebel CRM Event Publication and Subscription allows for easy subscriptions to consume event messages available in Kafka, these messages can be in Kafka from either of the following:

- Events in other Siebel external systems.
- Siebel itself as part of the first step in 2-Step or asynchronous publish.

You can then process the information received in Siebel and if required, publish necessary related events for downstream systems. This section is a part of *Siebel CRM Event Publication and Subscription*.

Using Partitions in Kafka with Siebel CRM Event Publication and Subscription

You can configure Siebel CRM Event Publication and Subscription to use Kafka partitions to improve scalability and enhance the speed of data exchange between Siebel CRM and other integrated applications through parallelism. When you use Kafka partitions (each partition is a sequential, ordered record of messages), the topic data is written to all the partitions specified for that topic. The messages within a topic are available in a number of partitions across multiple Kafka servers allowing:

- Multiple producer threads to write topic data to multiple Kafka partitions.
- Multiple consumer threads to read topic data from multiple partitions at the same time.

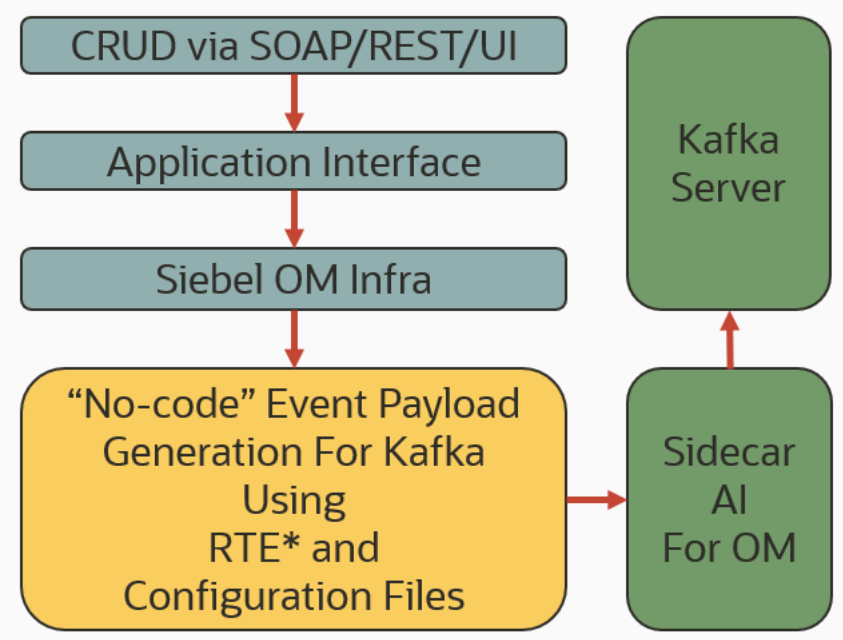
The AI Sidecars run the producer and consumer threads/applications inside them to integrate with Kafka. The number and specifications of producer and consumer threads running in the AI sidecars determine the latency and throughput of Siebel-Kafka message exchanges; hence their numbers should be planned carefully. For more information, refer the table *Planning Number of Consumers and Partitions*.

Using partitions with Siebel CRM Event Publication and Subscription is optional. But if you chose to use partitions, you must configure appropriate parameters in the configuration files (aieventconfig.txt and eventpayloadconfig.txt) and also plan Kafka configuration accordingly. For more information, see *Configuration Files for Payload Generation* and *Configuration Files for OM-AI Integration*.

Configuring Synchronous Publishing

You can configure synchronous event publishing using one of the following ways:

- **Option 1**: Publish with configuration only, without writing any custom code. The following represents conceptual data flow for a No-Code/1-Step Publish:



Following are the configurations required:

a. Create a business service based on *CSSEAIEventHandler* or its subclass.
b. Create RTE with Action Set as follows:

Attribute	Value
Action Type	Business Service
Business Service Name	MyDerivedAccountEvtPubSub
Business Service Method	SendEvent

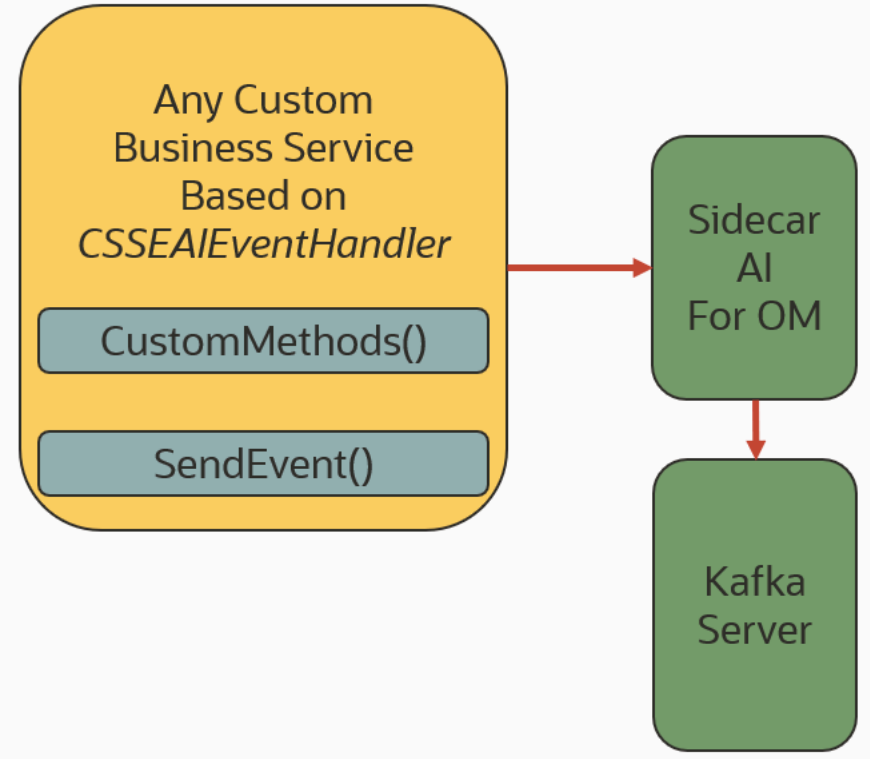
Attribute	Value

c. Configure eventpayloadconfig.txt, which involves adding entries in the eventpayloadconfig.txt file for the business components and fields whose data then get posted to Kafka. See *Configuration Files for Payload Generation*

Note: Siebel pre-delete record RTE is supported only for synchronous (or 1-Step) payload transmission to Kafka.

- **Option 2**: Send bespoke content to Kafka by implementing custom business logic.

  The following represents conceptual data flow for publishing any desired content to Kafka from custom implementation:



  Following are the configurations required:

  a. Create Business Service based on *CSSEAIEventHandler* or its subclass.
  b. Generate custom payload inside the custom business service created in the last step.
  c. Invoke *SendEvent* method.

  Note: In this option, you do not need to configure eventpayloadconfig.txt.

Enabling Event Subscription Screen

To enable event subscription screen

Note: This is a one-time activity.

1. Go to **Sitemap** > **Administration-Application Views**.
2. Add a view by the name **External Event** View
3. Assign users with appropriate responsibilities.
4. Clear the cache, log out and log in again.
5. You can now see the **Event Subscription** view.

Configuring Event Subscription

To configure a subscription in Siebel CRM

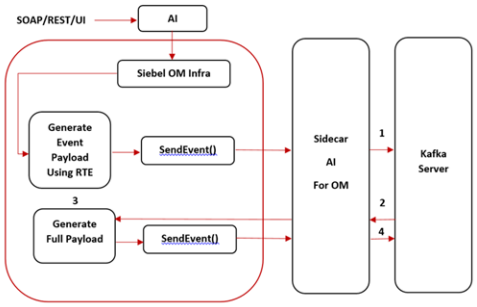
1. Go to **Sitemap > Administration-Application**.
2. Click **Event Subscription** and enter the following details in the view that opens:
   a. Business Service of choice under the field *Service* (any business service to process records received from Kafka).
   b. Business Service Method under the field *Method* (to process records received from Kafka).
   c. Topic name under the field *EventStore* (Siebel CRM subscribes to this topic in Kafka for event records).
   d. (Optional) Comma separated partition numbers under the field *Partitions*. When the partitions are specified for the topic (Event Store), the system invokes the Business Service and Method for the specified partitions of the topic.

Note:

a. If the Partitions field is not configured (left empty) for a topic (Event Store), the Business Service and Method will be invoked for all partitions of the topic configured in the **aieventconfig.txt** configuration file.
b. You must restart the AI sidecars after altering the number of partitions of a topic.

Information Flow in Asynchronous Publishing

The following flow explains the steps in the previous section.



In asynchronous or 2-Step publish, broadly the following sequence of information flow takes place:

1. The record ID (or any other identifier field data) is published to Kafka.
2. The published ID is then asynchronously consumed by Siebel using Subscription.

3. Siebel generates the complete payload per the eventpayloadconfig.txt file.
4. Siebel sends the complete payload to Kafka.

Therefore, configurations for asynchronous or 2-Step publish involves configurations for the following:

1. Synchronous Publish of record id (or any other necessary and equivalent field) to a topic.

   Sample payload:

   - Use case (extract from eventpayloadconfig.txt):

     ```
     Event: testevent1
     Business Service: Event Pub1
     Business Service Method: SendEvent
     Business Object: Contact
     Primary Business Component: Contact
     Context Field: Id
     ```

   - Old Payload:

     ```
     {
     "BusinessObjName": "Contact",
     "Id": "88-328W54",
     "ToSubscriber": "True",
     "BusinessCompName": "Contact"
     }
     ```

   - New Optional Payload:

     ```
     {
     "Id": "88-328W54",
     "ToSubscriber": "True"
     }
     ```

2. Subscribing to the topic from the last step and calling necessary business services (derived from CSSEAIEventHandler class) for subsequent publish of full payload to a desires Kafka topic.

Summary of Event Publication and Subscription Setup Steps

Overall set of steps involved in configuring this feature are depicted here as ready reckoner. Details of these steps are either Siebel standard processes or are detailed in this document at various places. All steps may not be applicable for your configuration use case.

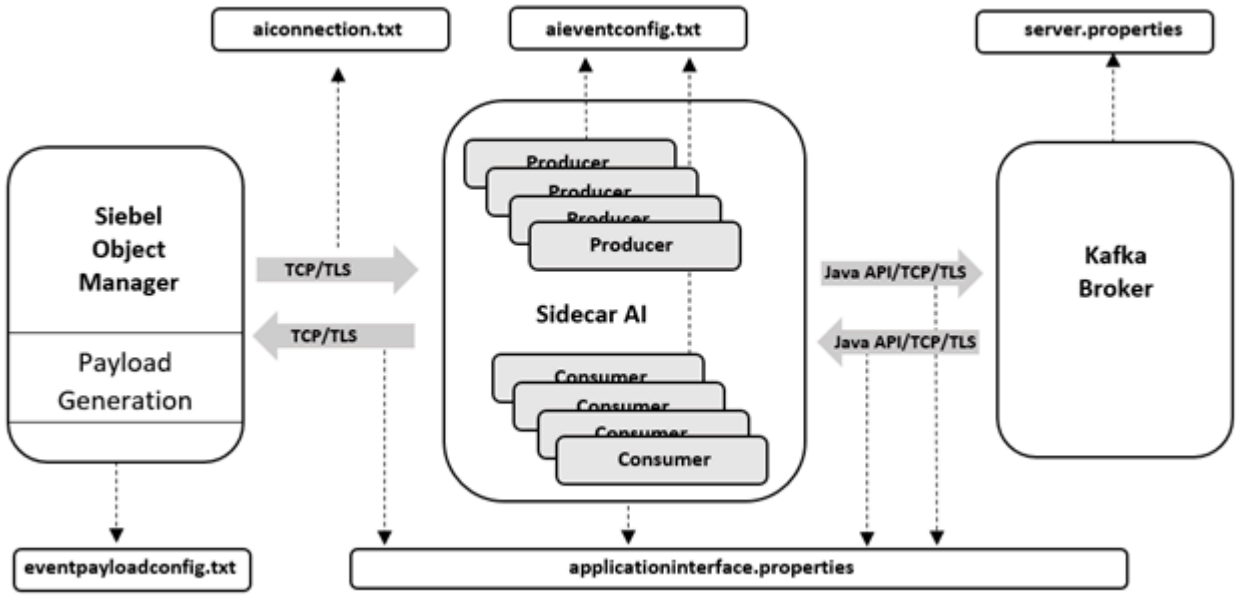
1. Enable component groups.
2. Add External Event View and assign responsibilities.
3. Create Business Services derived from Event Handler.
4. Create topics in Kafka, update server.properties files.
5. Create AI sidecars and configure application.properties, aieventconfig.txt and server.xml.
6. Inside Siebel Server directories, if not already present, create:
   - aiconnection directory and add file aiconnection.txt
   - eventconfig directory and add file eventpayloadconfig.txt
7. Set environment variables

8. Restart Siebel Servers, sidecar AIs and Kafka.
9. Configure Run-Time Events (RTE) and Event Subscriptions.

Configuration Files for Event Publication and Subscription

The following configuration files are required for event publication and subscription:

Location	Files
Siebel Server	**eventpayloadconfig.txt**: Used by object managers (OM) to generate the payload **aiconnection.txt**: Contains OM-AI sidecar connection information
AI Sidecar	**aieventconfig.txt**: For AI to propagate Siebel record to OM **applicationinterface.properties**: For Siebel side TLS configurations and other configurations
Kafka	**Kafka server.properties files**: SSL configs, Zookeeper connection info and other Kafka properties



This section contains the following topics:

- *Environment Variable Parameters for Enabling Event Publication and Subscription*
- *Configuration Files for Payload Generation*
- *Configuration Files for OM-AI Integration*

- *Publishing from Server Script*
- *Configurations for AI sidecar in Kafka Secure Communication*
- *Configurations for SASL Authentication*
- *Configurations for Using Multiple AI sidecars*

Environment Variable Parameters for Enabling Event Publication and Subscription

Following are the three Environment Variables associated with *Siebel CRM Event Publication and Subscription*:

Environment Variable	Value	Purpose
SIEBEL_EVENT_PUBSUB	1	If set then it will enable Event Pub-Sub Feature. If not set, then Event Pub-Sub Feature will not work. If Siebel Server and sidecar AI are on different machines, please ensure that it is set on both machines.
SIEBEL_EVENT_PUBSUB_TLS	1	If set then Siebel Server will have TLS connection with sidecar AI. If not set then Siebel Server will have TCP connection with sidecar AI. This needs to be set only for the machines hosting Siebel Server.
SIEBEL_EVENT_PUBSUB_PAYLOAD_STRUCT	1 or 2	Controls the payload structure (old/new) to be used to post messages to Kafka: If set to 1, uses the old payload structure. **Note:** When using: • RTE mechanism, you must not set the **`SingleRecordAsJSONObject`** parameter in the eventpayloadconfig.txt. Configuring **`SingleRecordAsJSONObject`** parameter in the **eventpayloadconfig.txt** will render all the configurations for the specific event (topic) invalid and no messages will be published to Kafka through the RTE trigger. • Custom Business Service implementation, the SingleRecordAsJSONObject property, if configured in the custom server scripts, will be ignored. If set to 2, uses the new payload structure.

Configuration Files for Payload Generation

The **eventpayloadconfig.txt** file defines the Siebel CRM fields whose contents are posted to Kafka. Optionally, you can also specify the partitions to which the data should be posted. This file is located inside the eventconfig directory under `<SIEBSRVR_ROOT>`.

Example path, `<.....>/ses/siebsrvr/eventconfig/eventpayloadconfig.txt`

In the context of entries in this file,

- **Event** represents the topic to publish to. An alternative name **Event Store** has been introduced for this parameter in the Siebel 24.12 release.

  > **Note:** For now, both the parameter names Event and Event Store are supported and can be used interchangeably. However, only the parameter name Event Store will be supported in the future releases. Hence, we recommend you to transition from Event to Event Store in the configuration at the earliest.

- (Optional) **Partitions** represents the Kafka partition numbers to write the topic. This parameter can take comma separated integer values.

  > **Note:** The Partitions parameter must be configured immediately after the Event/Event Store parameter. Configuring the Partitions parameter elsewhere, will render the payload invalid.

- Settings Fields: All and **All Except** reduce data entry.
- **Context Field** is used only for 2-Step or Asynchronous publish.
- (Optional)**SingleRecordAsJSONObject** is only applicable to the new JSON payload structure for RTEs, and for server scripts. It determines how a single data record of a child business component is posted. In the RTE mechanism of publishing, it is applicable only to the child business component and must be configured after the **Fields** parameter of the child business component. You can assign one of the following values to this parameter:
  - `true`: Posts a single data record of the child business object as a JSON object.
  - `false`: Posts a single data record of the child business object as an array. This is the default value.

> **Note:** You can use the `setProperty` method to configure the **SingleRecordAsJSONObject** parameter for a child Business Object in custom Business Service implementations through server script as follows:
>
> ```
> ChildBCPropertySet.SetProperty("SingleRecordAsJSONObject","<true/false>");
> ```

Siebel Event Publication and Subscription publishes messages based on the parameter configurations in the **eventpayloadconfig.txt** as follows:

- If the **Partitions** parameter is configured for a topic, topic messages are published to the specified partitions.
- If the **Partitions** parameter is not configured for a topic, topic messages are published based on the Kafka configurations.
- If the topic configured through the **Event Store** parameter, is not available in Kafka:
  - Kafka may automatically create the topic for Kafka brokers based on the Kafka broker properties **auto.create.topics.enable**, **num.partitions**, and so on. A topic is created in Kafka when the first message for the topic is posted from Siebel CRM to Kafka.

- Messages from Siebel CRM are posted to the partitions numbers configured for the topic in the **eventpayloadconfig.txt** file.

> **Note:** When using Avro serialization, you must ensure that the configuration of the **schemamapping** section is consistent with the topic and partition definitions in the **eventpayloadconfig.txt** file.

Structure of eventpayloadconfig.txt for synchronous publishing

Since RTE (Run-Time Events) configurations will refer to unique Business Service names, configuring different base business components (BC) will require specific business services based on CSSEAIEventHandler or classes derived from it. For example, Account BC can use *AccountEventGenBS*, Contact BC can use *ContactEventGenBS.* and so on.

Using `Fields:All` sends data in all active fields of the Business Component to Kafka. Using `Fields: All Except` sends all active fields other than those following these words to Kafka. For MVFs (Multivalue Field), child Business Component (on which the MVL is configured for the MVF) fields need to be added.

This is list of fields to be mentioned in aieventpayloadconfig.txt file for synchronous publish.

Field	Description
Event/Event Store	Name of the Kafka topic to publish to
Partitions	Lists the Kafka partitions to which the topic should be published to
Business Service	A business based on class CSSEAIEventHandler or classes derived from it
Business Service Method	Business service method name for publishing: SendEvent
Business Object	Business object name, for example, Contact
Primary Business Component	Event data of the primary component name to be published to Kafka
Fields	List of active fields of primary business component, separated by commas, to be included in the payload published to Kafka
Child Business Component	Necessary child component name that you want to publish data of
Fields	List of active fields of the child business component, separated by commas, to be included in payload published to Kafka
SingleRecordAsJSONObject	Determines if a single data record of a child business component is posted as a JSON object or as an array

> **Note:** There is no **Context Field** for synchronous publish. The Order of fields needs to be maintained as above.

Samples of eventpayloadconfig.txt for synchronous publishing:

```
Event Store: ContactEvent
Partitions: 0, 1, 2
Business Service: ContactEventSvc
Business Service Method: SendEvent
Business Object: Contact
Primary Business Component: Contact
Fields: First Name, Last Name, Contact Id
Child Business Component: Account
Fields: Name, Location, CSN, City, City State
SingleRecordAsJSONObject: true
Child Business Component: Action
```

```
Fields: Status, Activity, Type, End, Due, Owner, Start
SingleRecordAsJSONObject: false
 Child Business Component: Opportunity
Fields: Id
```

In the above sample file for synchronous publish, how the single data record of the child business components (Account and Opportunity) are posted is determined by the value of the environment variable `SIEBEL_EVENT_PUBSUB_PAYLOAD_STRUCT` and the `SingleRecordAsJSONObject` property of the child business components: In this case, if the value of `SIEBEL_EVENT_PUBSUB_PAYLOAD_STRUCT` is set to:

- **1 or is left empty**: The configuration for the topic "ContactEvent" is rendered invalid as the `SingleRecordAsJSONObject` parameter is configured for the child objects. Siebel Event Publication and Subscription will not perform the required publish process and will not post any ContactEvent messages to Kafka through the RTE trigger.
- **2**: The value of the `SingleRecordAsJSONObject` parameter is considered and the child business component is posted as follows:
-

```
{
 "Contact":{
 "Contact Id":"",
 "First Name":"test1",
 "Last Name":"outllok",
 "Account":{
 "Name":"A. K. Parker Inc.",
 "Location":"HQ-Corporate",
 "CSN":"10-103",
 "City":"Menlo Park",
 "City State":"Menlo Park, CA"
 },
 ……..
 ……..
 "Action": [
 {
 "Status":"Unscheduled",
 "Type":"Appointment",
 "Due":"04/19/2024 01:36:06",
 "Start":"",
 "Owner":"",
 "End":"",
 "Activity":""
 }
 ]
 ……..
 ……..
 "Opportunity": [
 {
 "Id":"A567"
 }
 ]
 ……..
 ……..
 }
}
```

  Observe that the:

  - Account child business component is posted as a JSON Object as the `SingleRecordAsJSONObject` property for it is set to `true`.
  - Action child business component is posted as an array as the `SingleRecordAsJSONObject` property for it is set to `false`.

- Opportunity child business component is posted as an array as the `SingleRecordAsJSONObject` property for it is not set.

Note: More than one data records are always posted as an array.

Sample of eventpayloadconfig.txt for asynchronous publishing:

```
Event:AccountSiebelInternalEvent

Business Service: AccountContextIdPubSvc

Business Service Method: SendEvent

Business Object:Account

Primary Business Component : Account

Context Field:Id

Event:AccountEvent

Business Service: AccountEventPubSvc

Business Service Method: SendEvent

Business Object:Account

Primary Business Component : Account

Fields:All

Child Business Component:Contact

Fields:FirstName
```

In the above sample file for asynchronous publish, the section that appears up to and including Context Field is responsible for publishing a record identifier, for example, an ID (ROW_ID of a Siebel CRM record) to the topic AccountSiebelInternalEvent. This publish is performed by the Send Event method of Business Service AccountContextIdPubSvc in the above example, where AccountContextIdPubSvc must be having CSSEAIEventHandler or a subclass as its class. Subsequently, a Business Service AccountEventPubSvc (also derived from CSSEAIEventHandler class or its subclass) will be publishing the whole payload to the topic AccountEvent. AccountEventPubSvc must be configured in the Event Subscription view to subscribe to the topic AccountSiebelInternalEvent (where Siebel ROW_ID was published) and call its Send Event method. Because subscription requires configuration in aieventconfig.txt file as well, necessary entries must be made within Events array of aieventconfig.txt.

Configuration Files for OM-AI Integration

Two files are primarily involved in maintaining OM-AI Integration for Event Pub-Sub functionality. They are **aiconnection.txt** and **aieventconfig.txt.**

- **aiconnection.txt**: Defines which AI sidecar is serving which Object Manager component at which port. It is located under <SIEBSRVR_ROOT>, where you can create a directory named **aiconnection** if not present already.

  Example path, **<……>/ses/siebsrvr/aiconnection/aiconnection.txt.**

  It uses alias name of the component. Multiple instances of a component can be used with different AI sidecars.

  Sample structure

  <ComponentAlias>=Comma Separated List of <AI hostname which acts as sidecar to OM >:<port>

  Sample Data Entries

  EAIOBjMgr_Enu=*mydomain.mycompany.com*:9091;localhost:9092

  UCMObjMgr_enu=localhost:9093;*mydomain.mycompany.com*:9094

  SCCObjMgr_enu=*mydomain.mycompany.com*:9095;localhost:9096

  > **Note:** AI connect string info is read in round robin fashion. A connection is opened between a component (bearing its alias name) and its AI sidecar, data is sent and then the connection is closed.

- **aieventconfig.txt**: The configurations in aieventconfig.txt is relevant for AI sidecars to communicate with both Object Managers and Kafka. AI sidecar uses configuration in **aieventconfig.txt** to consume event records from Kafka and propagate (over SISNAPI) to Siebel Component.

  Create an AI (for example using Installer and SMC) to be used as sidecar for Event Publication and Subscription, then place **aieventconfig.txt** inside web apps directory of that sidecar AI.

  Example path, `<.....>/ses/ applicationcontainer_external_fins/webapps/aieventconfig.txt`

  > **Note:** Topics subscribed to must have entry within Events array in **aieventconfig.txt**.

  Additionally, these topics must be configured under Event Store field in Event Subscription screen. Topic name cannot be repeated for a given AI.

  Optionally, you can use partitions for a topic through the optional parameter **Partitions**. Siebel CRM Event Publication and Subscription uses the value or values of the **Partitions** parameter in the **aieventconfig.txt** file to determine the partitions to subscribe to for a topic, as follows:

  - If you configure the **Partitions** parameter for the topic, Siebel CRM Event Publication and Subscription will consume data only from those partitions for that topic.
  - If you create the topic with 'n' number of partitions but do not configure the **Partitions** parameter for the topic, Siebel CRM Event Publication and Subscription will set partitions for the topic to 0, 1,.., n-1 and will subscribe to all the partitions of the topic.
  - If you do not create a topic and do not configure the **Partitions** parameter, Siebel CRM Event Publication and Subscription will set the partitions for the topic to 0, 1, 2, ..., n-1. Here, 'n' is the default number of partitions set for the Kafka Broker through the **num.partitions** property in the **config/server.properties** file.

  Optionally, you can also configure the **schemamapping** section to use Avro serialization. The schema mapping information is used by both producer and the consumer, as follows:

  - The producer uses the Avro schema to create an Avro compliant JSON message and writes it to Kafka broker.
  - The consumer uses the Avro schema to convert a JSON message into a Siebel-compliant JSON format.

  If the producer and consumer are on different sidecars, you must add the same schema mapping information in both the sidecars. For more information, see *Avro Serialization in Siebel CRM Kafka Integration*

  An object manager is required for bootstrapping Event Publication and Subscription functionalities during start-up. This object manager is referred to as `BaseSiebelComponent` in aieventconfig.txt.

  Similarly, you must enter information about object managers that will process subscriptions within Siebel CRM – these object managers are referred to as **SiebelComponent** under the `Events` array to process subscription requests from specific event stores like Kafka topics. The `BaseSiebelComponent` is also used to process subscriptions when no object manager is defined as part of the information about **SiebelComponent** for a specific event store (for example, a Kafka topic).

  > **Note:** Only one `BaseSiebelComponent` must be defined whereas you can use as many **SiebelComponent** as required (one object manager can process subscriptions from more than one event store like topic).

  In aieventconfig.txt, therefore, the configurations of the object managers (`BaseSiebelComponent` and all the **SiebelComponent**) needs a user to be defined that has appropriate access permissions comprising of positions, responsibilities and any other configurations for processing data related to Event Publication and

Subscription feature. For example, these users must have access to all necessary Siebel CRM building blocks like screens, views, business objects, business components, business services, workflows, and others.

This table describes the list of aieventconfig.txt fields and sections.

Field Name	Parent	Description
BaseSiebelComponent	ConfigParam	Fetches the event subscription data during AI sidecar startup and processes the data from the create, update, and delete operations in the Event Subscription view
Alias	BaseSiebelComponent	Base component's alias
SessionPoolSize	BaseSiebelComponent	The maximum number of JDB sessions that the AI sidecar can create concurrently while fetching the Event Subscription view records.
User	BaseSiebelComponent	Login User ID to allow the base component to process request
Password	BaseSiebelComponent	Encrypted password according to security profile (DB/LDAP and so on)
SessionTime	BaseSiebelComponent	Base component's session timeout
DefaultKafkaConsumerPollInterval	BaseSiebelComponent	Default polling interval of Kafka consumer running in AI to poll for records from Kafka
NumberOfProducers	ConfigParam	Number of producers in AI sidecars
KafkaProducerMaxRequestSize	ConfigParam	Sets producer property max.request.size - the maximum size of a request in bytes
KafkaProducerPollInterval	ConfigParam	Poll interval (in milliseconds) at which producer running in AI sidecar writes to Kafka
KafkaConsumerMaxPartitionFetchBytes or KafkaPartitionConsumerFetchMaxBytes	ConfigParam	Used to set Kafka Consumer config max.partition.fetch.bytes. An alternative name "KafkaPartitionConsumerFetchMaxBytes" has been introduced for this parameter

Field Name	Parent	Description
		Note: For now, both "KafkaConsumerMaxPartitionFetchBytes" and "KafkaPartitionConsumerFetchMaxBytes" are supported and can be used interchangeably. Eventually, support for the parameter name "KafkaConsumerMaxPartitionFetchBytes" will be stopped, and only the parameter name "KafkaPartitionConsumerFetchMaxBytes" will be supported. We advise you to utilize the intermediate time to make the transition from "KafkaConsumerMaxPartitionFetchBytes" to "KafkaPartitionConsumerFetchMaxBytes" in the configuration.
KafkaConsumerFetchMaxBytes or KafkaTopicConsumerFetchMaxBytes	ConfigParam	Used to set Kafka Consumer Config fetch.max.bytes. An alternative name "KafkaTopicConsumerFetchMaxBytes" has been introduced for this parameter **Note**: For now, both "KafkaConsumerFetchMaxBytes" and "KafkaTopicConsumerFetchMaxBytes" are supported and can be used interchangeably. Eventually, support for the parameter name "KafkaConsumerFetchMaxBytes" will be stopped, and only the parameter name "KafkaTopicConsumerFetchMaxBytes" will be supported. We advise you to utilize the intermediate time to make the transition from "KafkaConsumerFetchMaxBytes" to "KafkaTopicConsumerFetchMaxBytes" in the configuration.
Events	ConfigParam	Array of Kafka Topics for which subscription has been configured. Empty if only 1-step is in scope or when no subscription is configured.
Topic or EventStore	Events	Name of the Kafka Topic that AI sidecar has subscribed to **Note**: For now, both "Topic" and "Event Store" are supported and can be used

Field Name	Parent	Description
		interchangeably. In the future releases, only the parameter name “EventStore” will be supported. Hence, we advise you to make the transition from “Topic” to “EventStore” in the configuration at the earliest.
Partitions	Events	Specify the partition number/partition numbers from which the consumers will consume data of a given topic.
KafkaConsumerGroupId	Events	Consumer group id for the topic
NumberOfConsumers	Events	Number of consumer threads running in AI sidecar to consume record from the subscribed topic.
KafkaConsumerPollInterval	Events	Poll interval (in milliseconds) at which Kafka consumer running in AI sidecar reads from Kafka topic.
SiebelComponent	Events	Section to detail the object manager component which will process the event data consumed by AI sidecar from Kafka topic.
Alias	SiebelComponent	Alias name of the object manager component which will process the event data consumed by AI sidecar from Kafka.
SessionPoolSize	SiebelComponent	The maximum number of JDB sessions that can be created concurrently while consuming events of a specific topic. **Note**: You must configure the value of this parameter based on the traffic expected for the topic.
User	SiebelComponent	User ID to login to component to process event data.
Password	SiebelComponent	Encrypted Password to login to component to process event data.

Field Name	Parent	Description
SessionTime	SiebelComponent	Component session timeout
schemamapping	ConfigParam	Optional. You must configure this section to enable Avro serialization. For more information, see *Avro Serialization in Siebel CRM Kafka Integration*.

Sample aieventconfig.txt file Entries:

```
{
 "ConfigParam" : {
 "BaseSiebelComponent" :{
 "Alias":"EAIObjMgr_enu",
 "SessionPoolSize": "5",
 "User":"SADMIN",
 "Password":"<EncryptedPassword>",
 "SessionTime":"200",
 "DefaultKafkaConsumerPollInterval":"100"},
 "NumberOfProducers": "5",
 "KafkaProducerMaxRequestSize": "20971520",
 "KafkaProducerPollInterval":"100",
 "KafkaConsumerMaxPartitionFetchBytes": "20971520",
 "KafkaConsumerFetchMaxBytes": "20971520",
"Events":
 [{ "EventStore":"contactevent",
 "Partitions": "0,1,4",
 "KafkaConsumerGroupId": ”<ConsumerGroupId>",
 "NumberOfConsumers": "5",
 "KafkaConsumerPollInterval": "100",
 "SiebelComponent": {
 "Alias":"SCCObjMgr_enu",
 "SessionPoolSize": "10",
 "User":"SADMIN",
```

```
 "Password":"<EncryptedPassword>",

 "SessionTime":"900"

 }}

 ]}

}
```

One can understand that the Events array includes information to process subscription, like which topic is being subscribed to, the number of partitions for the topic how many consumer threads/applications to run inside AI sidecar to process this subscription, polling interval of that topic, and after records are read by the consumers in AI, which object manager component to send to in order to further process the event records received.

> **Note:** The Business Service and Method entered in Event Subscription screen will be invoked to execute necessary business logic.

Therefore, when there is no subscription running, for example when only “1-Step” outbound publish only is in scope, Events array will be empty and denoted by [].

The number of consumers and partitions are tightly coupled with each other. To ensure efficient use of computing resources and best throughout, you must plan the number of consumers subscribing from partitions in the aieventconfig.txt file as follows:

Use Case	Example	Result
When the value of `NumberOfConsumers` is equal to the number of partitions for a topic, each consumer will consume data from only one partition.	"NumberOfConsumers": "4" "Partitions": "0,1,2,3"	Consumer-0 will consume data from partition 0 Consumer-1 will consume data from partition 1 Consumer-2 will consume data from partition 2 Consumer-3 will consume data from partition 3
When the value of `NumberOfConsumers` is set to 1 and number of partitions are more than for a topic, then the single consumer will consume data from all the partitions of that topic	"NumberOfConsumers": "1" "Partitions": "0,1,2,3"	Consumer-0 will consume data from partition 0 Consumer-0 will consume data from partition 1 Consumer-0 will consume data from partition 2 Consumer-0 will consume data from partition 3
When the value of `NumberOfConsumers` is less than the number of partitions to subscribe to for a topic, each consumer will assigned to one partition except for the last consumer. The last consumer will consume from the rest of the partitions and therefore	"NumberOfConsumers": "3" "Partitions": "0,1,2,3"	Consumer-0 will consume data from partition 0 Consumer-1 will consume data from partition 1 Consumer-2 will consume data from partitions 2,3

Use Case	Example	Result
might be slower in performance compared to the other consumers.		
When the value of **`NumberOfConsumers`** is greater than the number of partitions to subscribe to for a topic, the number of consumers started will be equal to the number of partitions.	"NumberOfConsumers": "4" "Partitions": "0,1,2"	Consumer-0 will consume data from partition 0 Consumer-1 will consume data from partition 1 Consumer-2 will consume data from partition 2 Consumer-3 will not be started

- **applicationinterface.properties**: Another file that needs to be configured inside AI sidecars is the applicationinterface.properties file. It is required for various configurations for communication between sidecar AI and Kafka such as enabling SSL, using OAuth 2.0 authentication, and so on.

  This table describes the additional properties that need to be appended to the existing properties present in applicationinterface.properties file of AI sidecars.

  Properties with names beginning with **Kafka** is used by AI sidecar to communicate with Kafka servers (brokers). Properties with names beginning with **AI** is used by AI sidecar to communicate with Siebel CRM object managers (session based interactive components).

Property	Description
TLSEnabledCipherSuites	Enabled cipher suites. The following cipher suites are supported: **`TLS_AES_128_GCM_SHA256, TLS_AES_256_GCM_SHA384`**
AIEgressPort	AI sidecar TCP Server Port
KafkaServers	List of Kafka brokers with ports, separated by commas> (IMPORTANT: For SSL, please use fully qualified hostname. For example: *server12.company.com*:9093. Not just server12 or localhost
SecureAIToKafkaCommunication	Enter value TRUE or FALSE
SecureAIEgressServer	Enter value TRUE or FALSE.To start TLS server on AI
AIEgressServerKeyStoreType	Enter the keystore type. For example: JKS
AIEgressServerKeyStoreName	Enter the Siebel server keystore name For example: **`Z:\\siebel\ \applicationcontainer_external\\siebelcerts\\siebelserverkeystore.jks>`**
AIEgressServerKeyStorePassword	Key Store encrypted Password. For example: **`cWwvMzZqOFVlN0grTEQ2M1VpeUNxUjlSelFldDhjWFFzbjRQcWpndnVBSm5oa`**

Property	Description
KafkaKeyStoreType	Keystore type. For example: JKS
KafkaKeyStoreName	Siebel client keystore name for AI-Kafka communication. For example:`Z:\\siebel\\applicationcontainer_external\\siebelcerts\\siebelclientkeystore.jks>`
KafkaKeyStorePassword	Key Store encrypted Password. For example: `cWwvMzZqOFVlN0grTEQ2M1VpeUNxUjlSelFldDhjWFFzbjRQcWpndn`
KafkaTrustStoreType	Enter the trust store type. For example: JKS
KafkaTrustStoreName	Kafka Trust Store location. For example: `Z:\\siebel\\applicationcontainer_external\\siebelcerts\\siebelkeystore.jks>`
KafkaTrustStorePassword	Trust Store encrypted Password. For example: `cWwvMzZqOFVlN0grTEQ2M1VpeUNxUjlSelFldDhjWFFzbjRQcWpndnVBSm5oaUErd`
KafkaPassword	Kafka user's encrypted password, to access the private key inside the keystore. It maps to the Kafka `ssl.key.password` in the kafka `server.properties`, `consumer.properties`, and `producer.properties` files.For example: `cWwvMzZqOFVlN0gr`.
KafkaOauthClientID	Required only for SASL/OAUTHBEARER configuration. The client ID from the OAuth 2.0 server for SASL/OAUTHBEARER configuration.
KafkaOauthScope	Required only for SASL/OAUTHBEARER configuration. The scope from the OAuth 2.0 server that limits access to specific resources.
KafkaOauthClientSecret	Required only for SASL/OAUTHBEARER configuration. The client secret (encrypted) from the OAuth 2.0 server. Note: You must encrypt the client secret from the OAuth server using the Siebel encryption utility (java -jar EncryptString.jar <password>) and assign the encrypted string as the value of this parameter.
KafkaOauthTrustStorePassword	Required only for SASL/OAUTHBEARER configuration. The truststore password (encrypted) that is used by the Kafka client to connect to the Oauth 2.0 server when using SSL. Note: You must encrypt the truststore password using the Siebel encryption utility (java -jar EncryptString.jar <password>) and assign the encrypted string as the value of this parameter.
KafkaOauthTrustStore	Required only for SASL/OAUTHBEARER configuration. The truststore (JKS) file that is used by the Kafka client to securely connect to the OAuth 2.0 server when using SSL.

Property	Description
KafkaOauthEndPointURL	Required only for SASL/OAUTHBEARER configuration. The OAuth token endpoint URL from the OAuth 2.0 server to retrieve tokens.

Publishing from Server Script

You can publish event data from Siebel CRM to Apache Kafka using server scripts. This section lists simplified examples to demonstrate how to send event data to Apache Kafka using a server script and the impact of the `SingleRecordAsJSONObject` property.

Note: The examples, given below, are a section of the server script and do not follow the best practices guidelines and do not include error handling, object dereferencing, etc. In real-world implementations, we recommend you follow the best practice guidelines and complete the server script before running it.

You use the `setProperty` method to publish a topic to specific Kafka partitions. The following example is a section of the server script that demonstrates how to publish event data to specific partitions:

```
function Service_PreInvokeMethod (MethodName, Inputs, Outputs)
{
 var nReturn = ContinueOperation;
 var oBS;
 var ch1;
 var outPS;
 var conPS;
 var accPS;

 switch (MethodName) {
 case "PublishContact":
 ch1 = TheApplication().NewPropertySet();
 outPS = TheApplication().NewPropertySet();
 conPS = TheApplication().NewPropertySet();
 accPS = TheApplication().NewPropertySet();
 accPS.SetProperty("topic","AccountData");
 accPS.SetProperty("partitions", "0,1,3");
 ch1.SetType("Contact");
 ch1.SetProperty("First Name", "Anil");
 ch1.SetProperty("Last Name", "Kumar")
 conPS.AddChild(ch1);
 conPS.SetProperty("Account Name","SF");
 conPS.SetType("Account");
 accPS.AddChild(conPS);
 oBS = TheApplication().GetService("Event Handler");
 oBS.InvokeMethod("SendEvent", accPS, outPS);
 nReturn = CancelOperation;
 break;
 }
 return nReturn;
}
```

In the example above, the **Partitions** property for AccountData is set to the values 0, 1, and 3. Hence, the JSON message generated for AccountData using this server script will be posted to these partitions.

The following example demonstrates how a JSON payload is posted to Kafka based on the value of the `SingleRecordAsJSONObject` property:

```
function Service_PreInvokeMethod (MethodName, Inputs, Outputs)
{
 var nReturn = CancelOperation;
 var oBS;
 var topPS;
 var outPS;
 var conPS;
 var accPS;
 var fieldList = ["First Name", "Last Name"];

 switch (MethodName) {
 case "PublishContact":
 topPS = TheApplication().NewPropertySet();
 outPS = TheApplication().NewPropertySet();
 conPS = TheApplication().NewPropertySet();
 accPS = TheApplication().NewPropertySet();

 topPS.SetProperty("topic","customcontactevent");
 accPS.SetType("Account");
 accPS.SetProperty("Name",”NoAccount123");

 accPS.SetProperty("SingleRecordAsJSONObject",”true");

 conPS.SetType("Contact");
 for (var j=0; j<fieldList.length; j++)
 {
 conPS.SetProperty(fieldList[j], "NoName");
 }
 accPS.AddChild(conPS);
 topPS.AddChild(accPS);

 oBS = TheApplication().GetService("Event Handler");
 oBS.InvokeMethod("SendEvent", topPS, outPS);
 break;
 }
 return nReturn;
 }
 }
```

Based on the configuration in the above example and assuming that `SIEBEL_EVENT_PUBSUB_PAYLOAD_STRUCT` is set to `2`, the business objects are posted as follows:

```
{
 "Account": {
 "Name": "NoAccount123",
 "Contact": [
 {
 "First Name": "NoName",
 "Last Name": "NoName"
 }
 ]
 }
}
```

- A single data record of “Contact” will be posted as an array as the `SingleRecordAsJSONObject` property is not set.
- A single data record of “Account” will be posted as a JSON object as the `SingleRecordAsJSONObject` property is set to `true`.

Configurations for AI sidecar in Kafka Secure Communication

This section describes the configurations required for secure (SSL enabled) communication between OM and AI, and between AI and Kafka server. For more information see,*Configuration Files for OM-AI Integration*.

The Siebel CRM Application Interface communicates with Kafka through Kafka Client Java APIs over TCP/TLS and not using/over REST/HTTPs. Details about the latest version of those APIs (this version may or may not match with the version implemented/shipped with Siebel CRM) can be found at *https://kafka.apache.org/documentation/#api*.

For secure communication, only SSL security protocol is supported for authentication in the communication between Kafka brokers and Siebel CRM. You must use two separate certificates—one for server authentication and another for client authentication. This approach improves security and supports compliance. Refer to Kafka documentation for more details of all possible security protocols *https://kafka.apache.org/documentation/#security_overview*

In applicationinterface.properties file:

- The SecureAIEgressServer, SecureAIEgressServer properties should be set to TRUE or true.
- In KafkaServers property, fully qualified hostname to be used along with port number For example: *myserver.mycompany.com*. Do not use just myserver or localhost.
- Ensure that the following properties are set correctly:
  - AIEgressPort
  - KafkaKeyStoreName
  - KafkaKeyStoreType
  - KafkaKeyStorePassword
  - KafkaTrustStoreName
  - KafkaTrustStoreType
  - KafkaTrustStorePassword
  - KafkaPassword

In Kafka server (broker):

For configuration suitable for your business and performance use case, refer to:

https://kafka.apache.org/documentation/#brokerconfigs

Some of the parameters that may need changes:

- listeners=[list of comma separated listeners in the format `listener_name://host_name:port]` for example:
  ```
  listener_name://host_name:port] for example:
   listeners=PLAINTEXT://<Fully qualified hostname>:9092,SSL://<Fully qualified hostname>:9093
   Remove PLAINTEXT if only SSL is required
  ```
- `advertised.listeners=PLAINTEXT://<Fully qualified hostname>:9092,SSL://<Fully qualified hostname >:9093`
- `ssl.keystore.location=<Siebel server keystore location>`
- `ssl.keystore.password=<Siebel server keystore password>`

- `ssl.truststore.location=<truststorelocation>`
- `ssl.truststore.password=<trustore password>`
- `ssl.key.password=<pwd_value>`
- `ssl.client.auth=required`
- `zookeeper.connect=[List of Zookeeper Servers with port] for example <Fully qualified hostname>:2181` this is for Kafka to connect to zookeeper
- `message.max.bytes=20971520` Up to 5 MB is supported by Siebel CRM Event Publication Subscription.

Configurations for SASL Authentication

Siebel CRM-Kafka integration supports the following SASL authentication mechanisms:

- SASL/PLAIN
- SASL/SCRAM-SHA-256
- SASL/SCRAM-SHA-512
- OAUTHBEARER

Configuring SASL/PLAIN

SASL/PLAIN is a simple username/password authentication mechanism that can be used with TLS for encryption to implement secure authentication.

Note: All username and passwords mentioned below are for sample purposes only.

Changes in Kafka broker:

1. Create a new java Authentication and Authorization Service (JAAS) file called `kafka-server-jaas.conf` with contents like the one below. You need to add the Kafka server users in this file.

```
KafkaServer {
 org.apache.kafka.common.security.plain.PlainLoginModule required
 username="admin"
 password="admin-secret"
 user_admin="admin-secret"
 user_alice="alice-secret"
 user_siebel="siebel-secret";
};
```

2. Pass the above mentioned jaas file as JVM parameter in Kafka broker as follows:-

   `Djava.security.auth.login.config=<pathto kafka_server_jaas.conf>`

   For example, in a Windows Kafka environment, it can be done as follows. For configuration for other platforms, please check Kafka official reference documentation.

   ```
   set KAFKA_OPTS=-Djava.security.auth.login.config=kafka-server-jaas.conf
   ```

3. Update the `server.properties` file of the Kafka broker as follows:

   ```
   listeners=SASL_SSL://serverurl.oraclevcn.com:9095
   security.inter.broker.protocol=SASL_SSL
   ```

```
sasl.mechanism.inter.broker.protocol=PLAIN
sasl.enabled.mechanisms=PLAIN
```

and also add (for SSL configurations)

```
ssl.keystore.location=C:\\DebugBuild\\certs\\siebelserverkeystore.jks
```

Note: Please refer the Kafka documentation for more information, *https://kafka.apache.org/documentation / #security_sasl_plain*

Changes in AI sidecar for SASL/PLAIN:

1. Generate the encrypted password for the passwords setup in the Kafka-server-jaas.conf file (shown above) by using the EncryptString jar file in `<AI server>\webapps\siebel\WEB-INF\lib`.
2. Update the following properties in `applicationinterface.properties`:

```
SecureAIToKafkaCommunication=true
KafkaServers=<Kafka host name>:<Kafka port number>
KafkaAuthenticationEnabled=true
KafkaAuthenticationMechanism=PLAIN
KafkaAuthenticationUser=<SASL user name created in the Kafka-server-jaas.conf file on Kafka server>
KafkaAuthenticationPassword=<Encrypted password for the above user from the kafka-server-jaas.conf file>
KafkaKeyStoreType=JKS
KafkaKeyStoreName=<Key Store location e.g. Z:\\siebel\\applicationcontainer_external\\siebelcerts\
\siebelclientkeystore.jks>
KafkaKeyStorePassword=<Encrypted Key Store Password>
KafkaTrustStoreType=JKS
KafkaTrustStoreName=< Encrypted Trust Store location for example, in Windows, Z:\\siebel\
\applicationcontainer_external\\siebelcerts\\siebelkeystore.jks>
KafkaTrustStorePassword=<Trust Store Password>
KafkaPassword=<Encrypted Kafka user password>
```

Configuring SASL/SCRAM-SHA-256 or SASL/SCRAM-SHA-512

Suggested configuration changes in Kafka server (broker):

For more information, refer to *Kafka Official Documentation*. Use the following guidance:

Note:

- To encrypt the passwords, use the `EncryptString` utility from the folder **<Application External>\webapps \siebel\WEB-INF\lib**

  To encrypt the password, run: `java -jar EncryptString.jar <password>` and use the resulting encrypted string.
- All properties beginning with Kafka are used for sidecar AI-Kafka communication.

Make the following changes in Kafka broker:

1. Create new users in Kafka by following the official Apache Kafka documentation.
   In Windows:

```
kafka-configs.bat --alter --add-config "SCRAM-SHA-256=[iterations=8192,password=admin-
secret],SCRAM-SHA-512=[password=admin-secret]" --entity-type users --entity-name admin --bootstrap-
server localhost:9092
```

   or

```
kafka-configs.bat --zookeeper localhost:2181 --alter --add-config "SCRAM-SHA-
256=[iterations=8192,password=admin-secret],SCRAM-SHA-512=[password=admin-secret]" --entity-type users
 --entity-name admin
```

2. Create a file called `kafka_server_jaas.conf` with the following contents:

```
KafkaServer {
 org.apache.kafka.common.security.scram.ScramLoginModule required
 username="admin"
 password="admin-secret";
};
```

3. Pass the JAAS config file location as JVM parameter to each Kafka broker:

```
-Djava.security.auth.login.config=/path to kafka_server_jaas.conf
```

4. Configure SASL port and SASL mechanisms in server.properties:

```
listeners=SASL_SSL://<fully qualified server name>:<port number, for example 9093>
security.inter.broker.protocol=SASL_SSL
sasl.mechanism.inter.broker.protocol=SCRAM-SHA-256 (or SCRAM-SHA-512)
sasl.jaas.config=org.apache.kafka.common.security.scram.ScramLoginModule require
d username=admin password=admin-secret;
sasl.enabled.mechanisms=SCRAM-SHA-256 (or SCRAM-SHA-512)
```

5. Configure SSL support configuration in server.properties:

```
ssl.keystore.location=<Siebel server keystore location>
ssl.keystore.password=<Siebel server keystore password>
ssl.truststore.location=<truststore location>
ssl.truststore.password=<truststore password>
ssl.key.password=<pwd_value>
ssl.client.auth=required
```

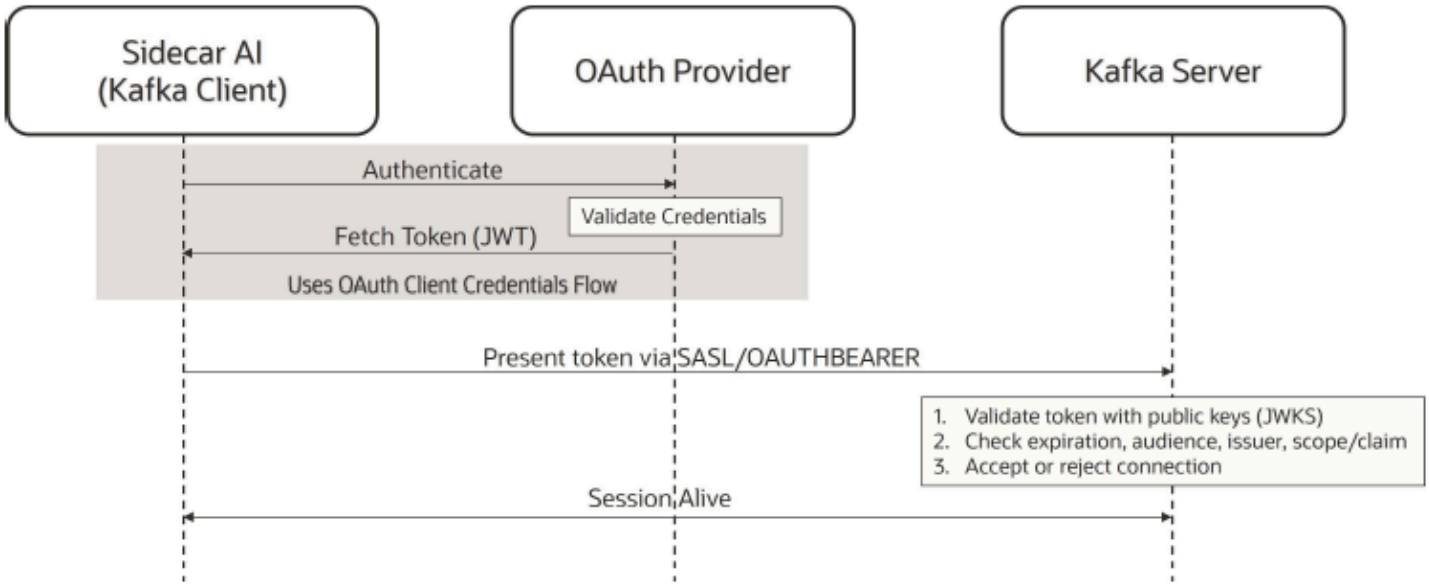
Corresponding configurations in applicationinterface.properties file of AI sidecars:

```
SecureAIToKafkaCommunication=true
KafkaServers=<Kafka host name>:<Kafka port number>
KafkaAuthenticationEnabled=true
KafkaAuthenticationMechanism=SCRAM-SHA-256 (or SCRAM-SHA-512)
KafkaAuthenticationUser=<SASL user name created in step 1 on Kafka server>
KafkaAuthenticationPassword=<Encrypted password for the above user created in step 1>.
KafkaKeyStoreType=JKS
KafkaKeyStoreName=<Siebel client keystore name>
KafkaKeyStorePassword=<Key Store Password>
KafkaTrustStoreType=JKS
KafkaTrustStoreName=<Trust Store location>
KafkaTrustStorePassword=<Trust Store Password>
KafkaPassword=<Kafka user password>
…
….other properties for Siebel-Kafka integration….
….other properties for AI not related to Siebel-Kafka Integration…
 …
```

Configuring SASL/OAUTHBEARER

In an Event Publication and Subscription setup, SASL/OAUTHBEARER is used to ensure secure and protected data exchange between Siebel CRM and Kafka. It enables Kafka clients (producers and consumers running in the sidecar AI) and brokers to authenticate with Kafka brokers using secure tokens issued by an OAuth 2.0 authorization server, such as Oracle IDCS, and so on.

A sample information exchange flow for SASL/OAUTHBEARER authentication is shown in the following diagram.



In this case, when a producer (Kafka client) running in the Sidecar AI wants to publish messages to Kafka, the following happens:

1. The producer fetches a token from the OAuth 2.0 server.
2. The producer connects to the Kafka Server (or broker) using the SASL/OAUTHBEARER mechanism and includes the token.
3. The Kafka Server validates the token with JWKS keys and verifies other authentication/authorization parameters.
4. Once validated, the Kafka Server initiates the secure event exchange with the producer over a session.

To configure Kafka with OAuth 2.0 authentication:

1. Retrieve the following information from the OAuth 2.0 server:

Data	Description	Sample data from the OAuth 2.0 server (Oracle IDCS)
Client ID	Unique identifier assigned to the client that is registered with an OAuth 2.0 server.	915cf6zzzz25413zzzzzz0897c754zzzz
Client secret	Password used to access protected resources.	xxxxxx-7eaf18zz-88x5-455x-8235-e846xxxx1111
Scope	Defines the level of access a client application has to a resource.	All
Token URL	URL that will be used to request an access token from the OAuth 2.0 server.	https:// xxxx.xxx.xxxx.xxx.com.com:443/oauth2/v1/token
JWKS URL	Endpoint URL that gives the list of the public keys used to verify the signature of JSON Web Tokens (JWT).	https://xxxx.xxx.xxxx.xxx.com:443/admin/v1/SigningCert/jwk

Data	Description	Sample data from the OAuth 2.0 server (Oracle IDCS)
Audience	Refers to the intended recipient of the access token.	abc
Issuer	Refers to the authorization server that created and signed the access token.	*https://identity.oraclecloud.com/*

2. Configure the Kafka broker to use OAuth 2.0 in the server.properties file. For more information, see *Changes in Kafka broker*.
3. Configure the sidecar AI to use OAuth 2.0 in the `applicationinterface.properties` file. For more information, see *Changes in Sidecar AI*.

Note: All username and passwords mentioned below are for sample purposes only.

Changes in Kafka broker:

To configure SASL/OAUTHBEARER with Event Publication and Subscription for Kafka broker:

1. Get the OAuth 2.0 server certificate. For example:
```
openssl s_client -showcerts -connect <OAuthServer:Port>
```
Note: The steps to generate the OAuth 2.0 certificate depends on the OAuth 2.0 server you are using. Please contact your OAuth 2.0 administrator to get the certificate.
2. Generate a truststore (JKS) file from the certificate.
```
"C:\Program Files\Java\jdk1.8.0_60\bin\keytool.exe" -importcert -trustcacerts -alias kafka-oauth-cert -
file oauth2Cert2.pem -keystore oauth2-openssl-truststore.jks -storepass changeit
```
3. Configure the JVM parameters using the truststore file and its password. For example, on Windows you can configure the JVM parameters by executing the following command at the command prompt where the Kafka server will be started:
```
set KAFKA_OPTS=-Djavax.net.ssl.trustStore=oauth2-openssl-truststore.jks -
Djavax.net.ssl.trustStorePassword=changeit
```
4. Update the OAuth2.0 properties in the `server.properties` file. Below is the sample of the `server.properties` file configured:

   Note: You must update the values of the properties in the server.properties file based on your environment set up.

   For SASL_PLAINTEXT:
```
advertised.listeners=SASL_PLAINTEXT://<fully qualified server name>:<port number, for example 9093>
listeners=SASL_PLAINTEXT://<fully qualified server name>:<port number, for example 9093>

sasl.enabled.mechanisms=OAUTHBEARER
listener.name.sasl_plaintext.sasl.enabled.mechanisms=OAUTHBEARER

listener.name.sasl_plaintext.oauthbearer.sasl.server.callback.handler.class=org.
apache.kafka.common.security.oauthbearer.OAuthBearerValidatorCallbackHandler
listener.name.sasl_plaintext.oauthbearer.sasl.login.callback.handler.class=org.
apache.kafka.common.security.oauthbearer.secured.OAuthBearerLoginCallbackHandler

listener.name.sasl_plaintext.oauthbearer.sasl.jaas.config=org.
```

```
apache.kafka.common.security.oauthbearer.OAuthBearerLoginModule
required clientId=<OAuth Client ID> clientSecret=<OAuth Client Secret> scope='xxxall';

listener.name.sasl_plaintext.sasl.oauthbearer.token.endpoint.url=https://<OAuth server name>:443/oauth2/
v1/token

listener.name.sasl_plaintext.sasl.oauthbearer.jwks.endpoint.url=https://<OAuth server name>:443/admin/
v1/SigningCert/jwk

listener.name.sasl_plaintext.sasl.oauthbearer.expected.audience=<recipient of the access token>
listener.name.sasl_plaintext.sasl.oauthbearer.expected.issuer=<server that issued the access token>


inter.broker.listener.name = SASL_PLAINTEXT
sasl.mechanism.inter.broker.protocol=OAUTHBEARER
```

For SASL_SSL:

```
advertised.listeners=SASL_SSL:// <fully qualified server name>:<port number, for example 9095>
listeners=SASL_SSL:// <fully qualified server name>:<port number, for example 9095>

sasl.enabled.mechanisms=OAUTHBEARER
listener.name.sasl_ssl.sasl.enabled.mechanisms=OAUTHBEARER

listener.name.sasl_ssl.oauthbearer.sasl.server.callback.handler.class=org.
apache.kafka.common.security.oauthbearer.OAuthBearerValidatorCallbackHandler
listener.name.sasl_ssl.oauthbearer.sasl.login.callback.handler.class=org.
apache.kafka.common.security.oauthbearer.secured.OAuthBearerLoginCallbackHandler

listener.name.sasl_ssl.oauthbearer.sasl.jaas.config=org.
apache.kafka.common.security.oauthbearer.OAuthBearerLoginModule required clientId=<OAuth Client ID>
 clientSecret=<OAuth Client Secret> scope='xxxall';

listener.name.sasl_ssl.sasl.oauthbearer.token.endpoint.url=https://<OAuth server name>:443/oauth2/v1/
token

listener.name.sasl_ssl.sasl.oauthbearer.jwks.endpoint.url=https://<OAuth server name>:443/admin/v1/
SigningCert/jwk

listener.name.sasl_ssl.sasl.oauthbearer.expected.audience=abc
listener.name.sasl_ssl.sasl.oauthbearer.expected.issuer=<server that issued the access token>

inter.broker.listener.name = SASL_SSL
sasl.mechanism.inter.broker.protocol=OAUTHBEARER

ssl.keystore.location==<Siebel server keystore location>
ssl.keystore.password=<Siebel server keystore password>
ssl.truststore.location==<truststore location>
ssl.truststore.password==<truststore password>
ssl.key.password==<pwd_value>
ssl.client.auth=required
```

Changes in Sidecar AI:

Configure SASL/OAUTHBEARER for sidecar AI in the in the `applicationinterface.properties` file. Here is a sample of the configured `applicationinterface.properties` file:

For SASL_PLAINTEXT:

```
KafkaServers=<Kafka host name>:<Kafka port number>
SecureAIToKafkaCommunication=false
KafkaAuthenticationEnabled=true
KafkaAuthenticationMechanism=OAUTHBEARER
```

```
KafkaOauthClientID=<OAuth client ID>
KafkaOauthClientSecret=<OAuth client secret>
KafkaOauthScope=xxxall
KafkaOauthEndPointURL=<OAuth server name>:443/oauth2/v1/token

KafkaOauthTrustStore=<Complete path of the OAuth truststore JKS file>
KafkaOauthTrustStorePassword=<OAuth truststore password>
```

For SASL_SSL:

```
KafkaServers=<kafka host name>:<kafka port number>
SecureAIToKafkaCommunication=true
KafkaAuthenticationEnabled=true
KafkaAuthenticationMechanism=OAUTHBEARER

KafkaOauthClientID=<OAuth client ID>
KafkaOauthScope=xxxall
KafkaOauthEndPointURL=<OAuth server name>:443/oauth2/v1/token

KafkaOauthTrustStore==<Complete path of the OAuth truststore JKS file>
KafkaOauthTrustStorePassword=<OAuth truststore password>

KafkaKeyStoreType=JKS
KafkaKeyStoreName=<Siebel client keystore name>
KafkaTrustStoreType=JKS
KafkaTrustStoreName=<truststore location>
KafkaPassword=<kafka user password >
```

For the parameter details, see the `applicationinterface.properties` parameter table. You must contact the OAuth 2.0 server administrator to get the values of the parameters.

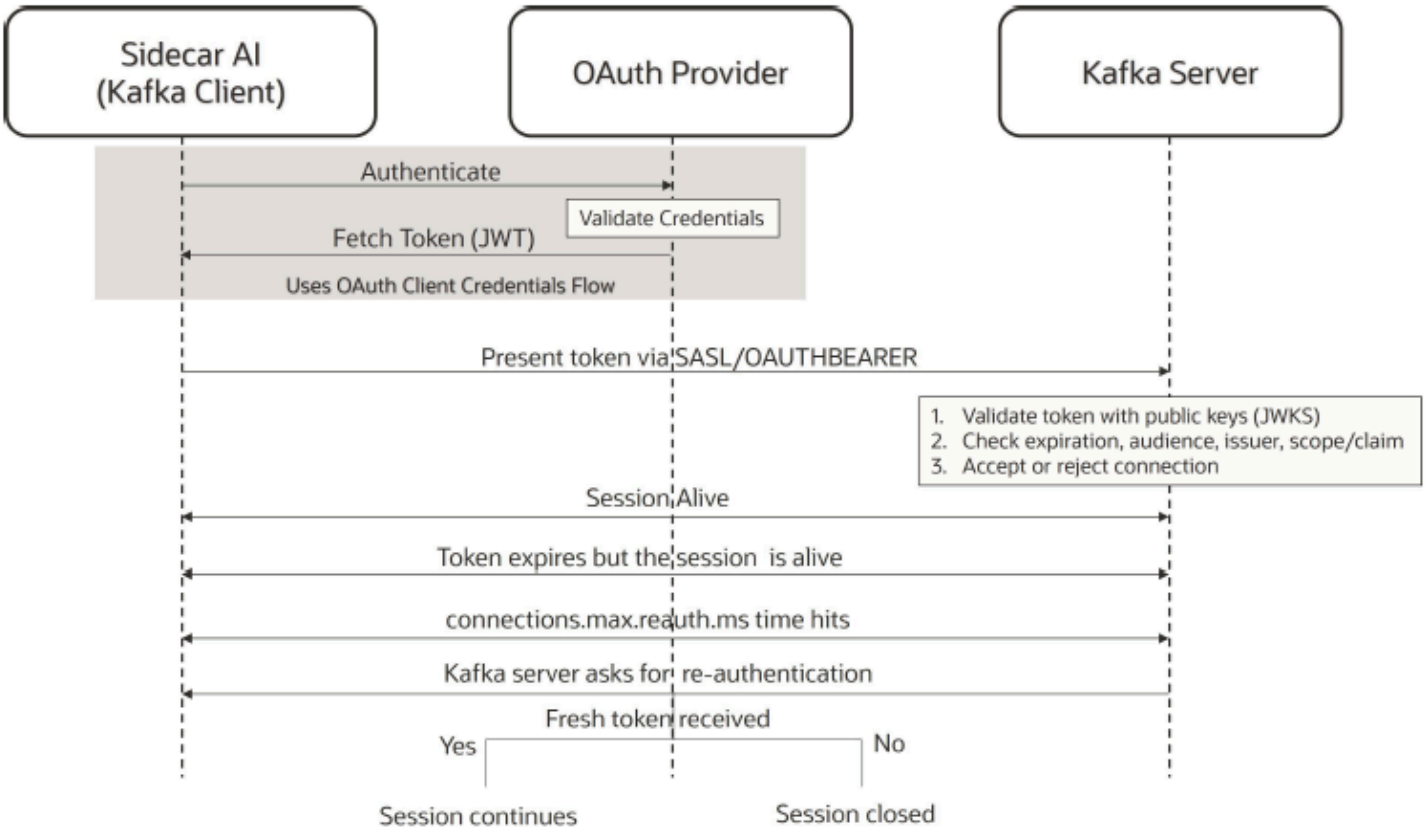
Note: The JWKS URL provided by the OAuth 2.0 server must have access from Kafka brokers. If the JWKS URL is not accessible from the Kafka broker, then the Kafka broker will not start. To verify if the JWKS URL is accessible to the user, open the JWKS URL in the browser. If the response is:

- An authentication error, it implies that the JWKS URL is not accessible.
- JWKS data in JSON format, it implies that the JWKS URL is accessible.

If your OAuth 2.0 authorization server is Oracle IDCS, you can configure access by Kafka brokers as follows:

1. Log in to the IDCS Administration Console.
2. Go to **Identity** > **Domains** > **Default domain** > **Settings** > **Domain settings**.
3. Under **Access signing certificate**, select **Configure client access**.

OAuth token expiration management:



By default, Kafka disables reauthentication to avoid unnecessary system load. Once a token is authenticated, Kafka does not re-authenticate it based on the token's expiration time. Instead, you can control how frequently the token is re-authenticated through the Kafka broker `connections.max.reauth.ms` parameter. You must set the value of this property carefully to prevent potential abuse or system overload by managing how often re-authentication should happen.

Configurations for Using Multiple AI sidecars

To run one or more additional sidecar AI apart from the vanilla AI sidecar, follow the steps:

1. Use Siebel Installer and SMC, like they are used for traditional AIs, for installing and configuring AIs that are to be used specifically for Event Pub-Sub feature. Use distinct names.

   For development and proof-of-concept use cases, one may recursively copy an existing traditional AI and rename the target directory.
2. Make necessary changes for files in AI sidecars following instructions mentioned earlier. Make sure that all ports in `<Sidecar AI directory>\conf\server.xml` are different from all other AIs.

Log File for Event Publication and Subscription

A dedicated log file EventLogger is available for AI sidecar processes. This file can be used for debugging, alongside necessary log files in Siebel Server, AIs and Kafka.

OCI Streaming with Apache Kafka - Configurations for Integrating Siebel Event Publication and Subscription

To integrate OCI Streaming with Apache Kafka with Siebel Event Publication and Subscription, follow the steps below:

1. Log in to Oracle Cloud (*https://cloud.oracle.com/*).
2. Create stream pools in the correct compartment as per the required specifications.
3. In the context of OCI Streaming service with Apache Kafka, a stream is logically equivalent to a topic in Apache Kafka on premises. Create the required streams for event exchange with Siebel CRM in the stream pools created in Step 2. Additionally, create the following streams with a single partition: sblnot and sblpart.

   Note: The sblnot and sblpart are for system use and should not be used for any other purpose.
4. Click the **Stream Pool Details** and get to the **Kafka Connection Settings** section.
5. Make a note of the following details:
   a. Bootstrap Servers
   b. SASL Connections Strings
   c. Security Protocol
   d. Security Mechanism
6. Update the value of password (AUTH_TOKEN) with the user authentication token in the SASL Connection Strings.

   Note: You must generate the user authentication token in the OCI Console.
7. Encrypt the password (updated in step 7) from the SASL Connections Strings using the Siebel Encryption utility as follows:

   ```
   java -jar
    EncryptString.jar <password>
   ```
8. Update the values of the following parameters with the details (captured in step 5) from the stream pool in the **applicationinterface.properties** file of AI sidecars:

   ```
   SecureAIToKafkaCommunication=true
   KafkaServers=<Value of Bootstrap Servers>
   KafkaAuthenticationEnabled=true
   KafkaAuthenticationMechanism=<Value of Security Mechanism> KafkaAuthenticationUser=<Value of username
   from the SASL SASL Connections Strings>
   KafkaAuthenticationPassword=<Encrypted SASL Connections String password>
   ```
9. Restart necessary AI sidecars.

Unit Testing Event Publication and Subscription

After you configure Kafka, you need to do preliminary tests to check if the configurations are correct. Sidecar APIs implement consumer and producer application, but Kafka installations also come with simple Consumer and Producer console (command line/shell) applications. You can configure consumer.properties and producer.properties file in Kafka installation, start necessary topics, send a request in the producer console and verify that it appears in the consumer console apps. These tests can ensure that various configurations (for example, SSL configurations) in Kafka and Siebel CRM are correct.

To Test Kafka producer and consumer console test apps

1. The following changes are to be made in consumer.properties and producer.properties of Kafka for testing communication over SSL:

```
security.protocol=SSL
ssl.keystore.location=<Fully qualified Siebel client keystore file location>
ssl.keystore.password=<Siebel client keystore password>
ssl.truststore.location=<Fully qualified Siebel Truststore file location> ssl.truststore.password=<pwd
 value>
ssl.key.password=<pwd value>
```

2. Start the Kafka consumer console app from the Kafka bin directory. For example:

```
kafka-console-consumer.bat --topic <name of topic> --bootstrap-server
<fully qualified Kafka server hostname>:9093 --consumer.config "D:\kafka\config\consumer.properties"
```

3. Start the Kafka producer console app from the Kafka bin directory. For example:

```
kafka-console-producer.bat --topic <name of topic> --bootstrap-server
<fully qualified Kafka server hostname>:9093 --producer.config "D:\kafka\config\producer.properties"
```

4. Test the communication between the producer and consumer by sending a sample JSON message from the producer and confirming that the consumer receives it. Ensure that the producer and consumer are on the same topic. For example:

   Sample command to send a message from the producer:

```
C:\kafka3\bin\windows>kafka-console-producer.bat --topic crm --bootstrap-server
 "phoenix680317.appsdev.fusionappsdphx1.oraclevcn.com":9095 --producer.config "C:\kafka3\config
\consumer.properties"
>{"name":"helloworld”}
```

   Sample command to verify the message received by the consumer:

```
C:\kafka3\bin\windows>kafka-console-consumer.bat --topic crm --bootstrap-server
 "phoenix680317.appsdev.fusionappsdphx1.oraclevcn.com":9095 --consumer.config "C:\kafka3\config
\consumer.properties"
{"name":"helloworld”}
```

   In the above example, the JSON payload sent from the producer to the `crm` topic must be displayed in the consumer console. This confirms that the Kafka configuration and setup is correct.

To configure SASL/OAUTHBEARER with Event Publication and Subscription for Kafka clients

1. Update the following properties in the `producer.properties` and `consumer.properties` files:

```
security.protocol=SASL_PLAINTEXT
sasl.mechanism=OAUTHBEARER

sasl.jaas.config=org.apache.kafka.common.security.oauthbearer.OAuthBearerLoginModule required \
clientId=<OAuth client ID> \
clientSecret=<OAuth client secret> \
scope="abcall";

sasl.oauthbearer.token.endpoint.url=<OAuth token URL>
sasl.login.callback.handler.class=org.apache.kafka.common.security.oauthbearer.OAuthBearerLoginCallbackHandler
```

2. Configure the JVM parameters using the truststore file and its password at the command prompt where the Kafka client will be started, as follows:

   On Windows:

```
set KAFKA_OPTS=-Djavax.net.ssl.trustStore=oauth2-openssl-truststore.jks -
Djavax.net.ssl.trustStorePassword=changeit
```

Kafka Consumer Application Scalability Guidelines

These are a few guidelines for Kafka consumer application scalability:

- Producer and consumer applications to subscribe to topics in Kafka from Siebel are implemented in AI sidecar. For higher throughput in event message processing and exchange, it is suggested that component cloning and AI sidecar replication are done. These in turn require careful considerations for scaling the number of consumers running inside AI sidecars.
- The total number of consumers for a given topic is equal to the total number of consumers running for the given topic in each AI side car and should be equal to the number of partitions of the given topic created inside Kafka. If the number of partitions is less than the total number of consumers for a given topic, some of the consumers will be unutilized.
- In a favorable situation, the total number of consumers for a given topic is equal to total of number of sessions created on components (for example, cloned telco OMs) which is going to process the consumed record. So, the number of component instances that should be running to handle the total number of consumers for a given topic should be decided depending on the number of sessions an instance of component can handle.

Avro Serialization in Siebel CRM Kafka Integration

This topic explains how to configure Avro serialization for Siebel CRM Kafka integration to optimize message exchange between the Kafka broker and client. It includes the following sections:

- *Creating Avro Schema Files*
- *Configuring Schema Mapping in aieventconfig.txt for Avro serialization*
- *Avro Schema Mapping Rules in Siebel CRM Event Publication and Subscription*

You can optimize message exchanges in Siebel CRM Kafka integration by using Avro serialization.

Avro is an open-source data serialization framework developed as part of the Apache Hadoop project and is widely used. It is the most common serialization framework used with Apache Kafka.

Avro relies on schemas. Avro schemas used during producing message and reading message from Kafka must be compatible. Avro also supports schema evolution, making it flexible as data requirements change. At present, Siebel CRM Event Pub/Sub does not support schema registry from Siebel CRM.

Using Avro serialization with Siebel CRM Kafka integration offers the following benefits:

- Compact size: Avro serialization produces smaller message payloads, which helps reduce storage and data transfer costs.
- Fast processing: Minimal overhead enables efficient, high-speed data processing.
- Schema consistency: Avro enforces schema consistency, which is ideal for systems that require data format standardization.
- Interoperability: Avro makes it easy for different producers and consumers to exchange data seamlessly.

Siebel CRM Kafka integration uses Avro version 1.9.2.

To implement Avro serialization in Siebel CRM Kafka integration, you must:

1. Create the Avro schema files. For more information, see *Creating Avro Schema Files*.
2. Configure the **schemamapping** section in the **aieventconfig.txt** file. For more information, see *Configuring Schema Mapping in aieventconfig.txt for Avro serialization*.

Creating Avro Schema Files

Payloads are generated internally within Siebel CRM Event Pub/Sub, just as they are when Avro serialization is not configured. Avro transformations are then applied according to the schemas, and the transformed and serialized messages are posted to Kafka.

Avro serialization supports various data types, for example string, record, array, and so on. Optionally, you can also specify default values for each field.

Avro fields in the schema file must follow these conventions:

- Field names must:
  - Start with a letter (A–Z or a–z) or an underscore (_).
  - Be followed by any combination of letters, digits (0–9), or underscores (_).
  - Not include hyphens (-), spaces, or special characters.
  - Examples:
    - Valid field names: userName, _id, age1
    - Invalid field names: 1name, user-name, user name
- Field values:
  - Strings must be UTF-8 encoded.

Mapping for Non-Avro Compliant Fields in Siebel JSON

Avro only supports field names that use the following characters: letters (a–z, A–Z), digits (0–9), and underscores (_). However, in Siebel CRM, some field names do not follow this format. For example, **Last Name** is a valid field name in Siebel CRM but is not compliant with Avro field naming rules.

To address this requirement, a field called **siebelName** has been introduced in the schema to support Siebel CRM fields that are not Avro-compliant. For example, if you want to include a field with a space, for example: **Last Name,** from Siebel CRM in your Avro JSON, you can add an element called **Last_Name** in the schema and set its **siebelName** property to **Last Name**. This approach allows you to map “Last Name” from the Siebel CRM payload to **Last_Name** in the Avro JSON, preserving the original value while maintaining Avro field naming conventions.

Note:

- If you do not configure the **siebelName** property for non-Avro compliant Siebel CRM field that you want to include, no transformation will occur.
- If the **siebelName** property is configured for a field, Kafka will scan the Siebel CRM payload (which is generated internally before transformation for Avro are applied) for a field which is mentioned as the value of **siebelName**. If the property is not configured, Kafka will search by the actual field name in the Siebel CRM payload.
- The Avro converted result will use the actual field name, not the **siebelName**.

Samples Avro Schema File and Output

Schema File 1	Output

<pre>{
"type": "record",
"name": "AccountCreation",
"fields": [
{
"name": "Account",
"type": {
"type": "record",
"name": "Account",
"fields": [
{
"name": "Name",
"type": "string"
},
{
"name": "CSN",
"type": "string"
},
{
"name": "Account_Holder",
"siebelName": "Contact",
"type": {
"type": "record",
"name": "Contact",
"fields": [
{
"name": "First_Name",
"type": "string",
"siebelName": "First Name"
},
{
"name": "Last_Name",
"type": "string",
"default": "NoName",
"siebelName": "Last Name"
}
]
}
}
]</pre>
	<pre>{
	"Account": {
	"Name": "Demo Account",
	"CSN": 8836,
	"Account_Holder": {
	"First_Name": "John",
	"Last_Name": "Doe"
	}
	}
	}</pre>

Schema File 1	Output
} }] }	

The above sample schema is configured to write information of newly created accounts and their associated contacts, with **Contact** defined as a child business component of the **Account**.

In the sample:

1. The **Contact** business component includes the following fields in the message posted to Kafka:
   - **First_Name,** with the **siebelName** property set to **First Name.**
   - **Last_Name,** with the **siebelName** property set to and **Last Name.**
2. The Siebel CRM payload scan is based on **First Name** and **Last Name**.
3. The Avro compliant output will have **First_Name** and **Last_Name**.
4. The **Account_Holder** field is the same as **Contact** field in the Siebel JSON.
5. The default value of **Last_Name** is set to **NoName**. If there is no match for this field (for example, when Last Name for the record in Siebel CRM is empty), the final output will default to **NoName**.

Schema File 2	Input	Output

{
"type": "record",
"name": "ContactAndKids",
"fields": [
{
"name": "MyContact",
"siebelName": "Contact",
"type": {
"type": "record",
"name": "Contact",
"fields": [
{
"name": "ContactId",
"type": "string"
},
{
"name": "First_Name",
"type": "string",
"siebelName": "First Name"
},
{
"name": "Last_Name",
"type": "string",
"siebelName": "Last Name"
},
{
"name": "Account",
"type": {
"type": "array",
"items": {
"type": "record",
"name": "Account",
"fields": [
{
	{
	"Contact": {
	"First Name": "Doe",
	"Last Name": "Jack",
	"Account": [
	{
	"Name": "HM",
	"Location": "HQ",
	"City": "Boston",
	"CSN": "1-12I9",
	"City State": "Boston, MA"
	}
	]
	}
	}
		{
		"MyContact": {
		"ContactId": "",
		"First_Name": "Doe",
		"Last_Name": "Jack",
		"Account": [
		{
		"Name": "HM",
		"Place": "HQ",
		"ID": "1-12I9",
		"Quote": [
		{
		"Name": ""
		}
		]
		}
		],
		"Opportunity": [
		{
		"OptyId": "",
		"Name": ""
		}
		]
		}
		}

Schema File 2	Input	Output

<pre>"name": "Name",
"type": "string"
},
{
"name": "Place",
"type": "string",
"siebelName": "Location"
},
{
"name": "ID",
"type": "string",
"siebelName": "CSN"
},
{
"name": "Quote",
"type": {
"type": "array",
"items": {
"type": "record",
"name": "Quote",
"fields": [
{
"name": "Name",
"type": "string"
}
]
}
}
}
]
}
}
},
{
"name": "Opportunity",
"type": {
"type": "array",
"items": {
"type": "record",
"name": "Opportunity",
"fields": [
{
"name": "OptyId",
"type": "string"
},
{
"name": "Name",
"type": "string",
"siebelName": "Name"
}
]
}
}
]
}
}
]
}</pre>		

The fields that are in the Siebel payload (generated internally before Avro transformations are applied according to the schema) will be populated with appropriate values and the fields that are not present in it will be left blank. In the above sample, the following fields that are not present in the Siebel CRM payload are left blank:

- Quote as a child of Account.
- Opportunity

Note: Avro serialization also supports the old Siebel payload format. In the old payload format for event publication and subscription, the primary business components and child business components are siblings in JSON. You must define your schema with this rule in mind. If you do not, the output will contain blank entries. For example:

Sample schema:

```
{
 "type": "record",
 "name": "ContactAndKids",
 "fields": [
 {
 "name": "MyContact",
 "siebelName": "Contact",
 "type": {
 "type": "record",
 "name": "Contact",
 "fields": [
 {
 "name": "ContactId",
 "type": "string",
 "siebelName": "Contact Id"
 },
 {
 "name": "First_Name",
 "type": "string",
 "siebelName": "First Name"
 },
 {
 "name": "Last_Name",
 "type": "string",
 "siebelName": "Last Name"
 },
 {
 "name": "Account",
 "type": {
 "type": "array",
 "items": {
 "type": "record",
 "name": "Account",
 "fields": [
 {
 "name": "Name",
 "type": "string"
 },
 {
 "name": "Place",
 "type": "string",
 "siebelName": "Location"
 },
 {
 "name": "ID",
 "type": "string",
 "siebelName": "CSN"
 }
```

```
      ]
      }
      }
      }

      ]
      }
      }
      ]
     }
```

Input Payload	Output Payload	Explanation

<pre>{
"Contact": {
"Fields": {
"Contact Id": "",
"First Name": "John",
"Last Name": "Dow"
},
"BusinessCompName": "Contact"
},
"Account": {
"Fields": {
"Name": "Hibbing Manufacturing"
},
"BusinessCompName": "Account"
},
"BusinessObjName": "Contact"
}</pre>
	<pre>{
	"MyContact": {
	"ContactId": "",
	"First_Name": "John",
	"Last_Name": "Dow",
	"Account": [
	{
	"Name": "",
	"Place": "",
	"ID": ""
	}
	]
	}
	}</pre>
		In this example, **Account** is defined as a child of **Contact** in the schema.
		
		However, in the old Siebel payload format, primary business components and child business components are represented as siblings in the JSON. As a result, the output contains blank entries for the **Account** fields.

Configuring Schema Mapping in aieventconfig.txt for Avro serialization

You must configure the **schemamapping** section in the **aiventconfig.txt** file to specify the topics, partitions, and schema file for each partition. Configuring partitions is optional, you can specify the schema file directly for the topic using the default parameter.

Sample schemamapping section in aieventconfig.txt

```
    schemamapping: [
     {
     "topic1":
     {
     "1": "topic1_1.txt",
     "2": "topic1_2.txt"
     }
     },
     {
     "topic2":
    {
     "default": "topic2_schema.txt"
     }
     }
     ]
```

In the example above:

1. Messages for **topic1** are produced to Kafka partitions 1 and 2.
2. Messages for **topic1 partition1** are serialized using the **topic1_1.txt schema file.**
3. Messages for **topic1 partition2** are serialized using the **topic1_2.txt schema file.**
4. Messages for **topic2** are produced to the Kafka topic using the **topic2_schema.txt schema file**.

Note:

- Ensure that the configuration of the **schemamapping** section is consistent with the topic and partition definitions in the **eventpayloadconfig.txt** file.
- When the default schema is configured for a topic, you must not add partitions for the topic. For example, you cannot configure as follows:

```
"schemamapping": [
 {
 "topic1": {
 "1": "topic1_1.txt",
 "default": "topic1_2.txt"
 }
 }
 ]
```

- Avro schema conversion for topic's messages will not occur if the topic is not included in the **schemamapping** section.

Avro Schema Mapping Rules in Siebel CRM Event Publication and Subscription

The section lists the Avro schema mapping rules in Siebel CRM Event Publication and Subscription with examples and explanations. This section includes the following:

- *General Schema Mapping Rule*
- *Data Type Schema Mapping Rules*
- *Rules for Duplicate Schema Fields in Payload*

General Schema Mapping Rule

This section explains the general Avro conversion rules for the following scenarios:

- Complete and partial matches
- Duplicates
- Duplicates with complete and partial matches
- Combining fields across different data structures
- Undefined fields

All examples in this section are based on the following sample schema:

```
{
```

```
 "type": "record",
 "name": "ContactAndKids",
 "fields": [
 {
 "name": "MyContact",
 "siebelName": "Contact",
 "type": {
 "type": "record",
 "name": "Contact",
 "fields": [
 {
 "name": "ContactId",
 "type": "string"
 },
 {
 "name": "First_Name",
 "type": "string",
 "siebelName": "First Name"
 }
 ]
 }
 }
 ]
}
```

- Complete and partial matches:
  - A complete path match occurs when the entire path defined in the schema is found at the root level of the input payload. For example:

Input Payload	Output Payload	Explanation
`{ "Contact": { "First Name": "Jack", "ContactId": "1234" } }`	`{ "MyContact": { "ContactId": "1234", "First_Name": "Jack" } }`	In this example, **Contact** is at the root level. This is considered as a complete path match, and the fields in the input payload are converted to Avro as follows: **Contact → ContactId to MyContact → ContactId** **Contact → First Name to MyContact → First_Name**

  - A partial path match occurs when the entire path defined in the schema is found under an element within the input payload, rather than at the root level. When processing an Avro schema, complete path match takes precedence over a partial path match. For example:

Input Payload	Output Payload	Explanation
`{ "Quote": { "Name": "qt1234", "Contact": { "First Name": "Jack", "Last Name": "Doe", "ContactId": "1234" } }`	`{ "MyContact": { "ContactId": "1234", "First_Name": "Jack" } }`	In this example, it is only a partial match because **Contact** is under **Quote**, not at the root level. The fields in the input payload are converted to Avro as follows: **Quote → Contact →ContactId to MyContact → ContactId**

Input Payload	Output Payload	Explanation
}		**Quote → Contact → First Name** to **MyContact → First_Name**

- Duplicates: A duplicate is defined as any case where scanning the payload returns multiple matches for all fields specified in the schema. For example:

Input Payload	Output Payload	Explanation
{ "Quote": { "Name": "qt1234", "Contact": { "First Name": "Jack", "ContactId": "1234" } }, "Communication": { "Name": "qt1234", "Contact": { "First Name": "Jill", "ContactId": "12345" } } }	{ "MyContact": { "ContactId": "", "First_Name": "" } }	In this example, the fields First Name and **ContactId** are found under both **Quotes → Contact** and **Communication → Contact** and are therefore considered duplicates. As a result, the output contains blank values for **First Name** and **ContactId** as there are duplicate records for these fields in the payload.
{ "Quote": { "Name": "qt1234", "Contact": { "City": "BLR" } }, "Communication": { "Name": "qt1234", "Contact": { "First Name": "Jill", "ContactId": "12345" } } }	{ "MyContact": { "ContactId": "12345", "First_Name": "Jill" } }	In this example, two instances of Contact are found under the elements Quote and Communication. However, the fields **First Name** and **ContactId** under **Contact,** as defined in the schema, are found only under, **Communication** > **Contact**. Therefore, this is not a case of duplicate records.

- Duplicates with complete and partial matches: In case of duplicates, the following rules apply:
  - A complete path match automatically takes precedence. For example:

Input Payload	Output Payload	Explanation
{ "Contact": { "First Name": "Jack", "ContactId": "1234" }, "Communication": { "Name": "qt1234", "Contact": { "First Name": "Jill",	{ "MyContact": { "ContactId": "1234", "First_Name": "Jack" } }	In this example, the fields **ContactID** and **First Name** are found both under the root-level **Contact** and under **Communication → Contact** and are therefore considered duplicates. However, the root-level **Contact** represents a complete path match

Input Payload	Output Payload	Explanation
"ContactId": "12345" } } }		according to the schema and takes precedence over partial path match (**Communication → Contact)**. As a result, the output sets the **Contact → First Name** to Jack and **Contact → ContactId** to 1234 from the root-level **Contact**.

- A complete path match also takes precedence over the type of the record. For this specific example, the below sample schema is used:

```
{
 "type": "record",
 "name": "ContactAndKids",
 "fields": [
 {
 "name": "MyContact",
 "siebelName": "Contact",
 "type": {
 "type": "record",
 "name": "Contact",
 "fields": [
 {
 "name": "ContactId",
 "type": "string"
 },
 {
 "name": "First_Name",
 "type": "string",
 "siebelName": "First Name"
 }
 {
 "name": "Account",
 "type": {
 "type": "record",
 "name": "Account",
 "fields": [
 {
 "name": "Name",
 "type": "string"
 }
 ]
 }
 }

 ]
 }
 }
 ]
}
```

Input Payload	Output Payload	Explanation

{
"Contact": {
"First Name": "Jack",
	{
	"MyContact": {
	"ContactId": "1234",
		In this example, the schema scans for **Contact** and an **Account** within

Input Payload	Output Payload	Explanation

<pre>"Last Name": "Doe",
"ContactId": "1234",
"Account": [
{
"Name": "Test Acc1",
"CSN": "88-36MIAD"
}
]
},
"Communication": {
"Name": "qt1234",
"Contact": {
"First Name": "Jill",
"ContactId": "12345",
"Account": {
"Name": "Test Acc12",
"CSN": "88-36MKAD"
}
}
}
}</pre>
	<pre>"First_Name": "Jack",
	"Account": {
	"Name": "Test Acc1"
	}
	}
	}</pre>
		Contact. Both **Contact** and **Account** are defined as records in the schema. In the input payload, the root-level **Contact** is a record, but **Account** inside it is a single element array. However, the Avro conversion selects the single-element array **Account** under root-level **Contact,** even though both **Contact** and **Contact → Account** under Communication->Contact are records.

- A complete path match will take precedence only if it includes at least one required field from the schema. If no required fields are present in the complete path match, then a partial path match that includes at least one required field will be considered instead. For example:

Input Payload	Output Payload	Explanation

<pre>{
"Contact": {
"First Name": "Jack"
},
"Communication": {
"Name": "qt1234",
"Contact": {
"First Name": "Jill",
"SingleRecordAsJSONObject":
"true",
"ContactId": "12345",
}
}
}</pre>
	<pre>{
	"MyContact": {
	"ContactId": "",
	"First_Name": "Jack"
	
	}
	}</pre>
		In this example, the root-level **Contact** only includes **First Name**. Since it is complete path match and includes one required field defined in the schema, **First Name** is set to **Jack** in the output. The scan ignores the partial path match **Communication → Contact,** even though it includes both **First Name** and **ContactId** from the schema.

- If a complete path match is not found, then an element with at least one non-duplicate field from the schema will be considered. If all the fields in a schema have duplicates, then empty values are returned for those fields. For example:

Input Payload	Output Payload	Explanation

<pre>{
"Quote": {
"Name": "qt1234",
"Contact": {
"First Name": "Jack",
"ContactId": "1234567"</pre>
	<pre>{
	"MyContact": {
	"ContactId": "",
	"First_Name": ""
	
	}</pre>
		In this example, all the fields are duplicated. As a result, the output contains only blank values.

Input Payload	Output Payload	Explanation

}
},
"Communication": {
"Name": "qt1234",
"Contact": {
"First Name": "Jill"
"ContactId": "1234567"
}
}
}
	}	

- Combining fields across different structures: Creating a match by combining schema fields across different data elements is not allowed. For example:

Input Payload	Output Payload	Explanation

{
"Quote": {
"Name": "qt1234",
"Contact": {
"First Name": "Jack"
}
},
"Communication": {
"Name": "qt1234",
"Contact": {
"ContactId": "12345"
}
}
}
	{
	"MyContact": {
	"ContactId": "12345",
	"First_Name": ""
	}
	}
		In this example, the system scans for the **First Name** and **ContactId** fields under **Contact**. However, First Name is available only under Quote→Contact and ContactId appears only under Communication→Contact. Fields from different elements, **Quote** and **Communication**, are not combined. As a result, the final output includes fields from only one data element. You must review and update the schema as needed.

- Undefined fields: The fields that are not defined in the schema are not converted and will not be included in the output payload. For example:

Input Payload	Output Payload	Explanation

{
"Contact": {
"First Name": "Jack",
"Last Name": "Dow",
"ContactId": "1234"
}
}
	{
	"MyContact": {
	"ContactId": "1234",
	"First_Name": "Jack"
	}
	}
		In this example, the input payload includes **Last Name** which is not defined in the schema. As a result, **Last Name** is not included in the output payload.

Data Type Schema Mapping Rules

The following table lists the data type rules for Avro conversion:

Schema Type	Input Payload Type	Output Payload Type	Explanation
Record	Record	Record	A record in the input payload that is defined as a record in the schema file is also treated as a record in the output payload.
Record	Single element array	Record	A single element array in the input payload is treated as a record in the output payload.
Record	Multi element array	Empty record	If an array contains multiple elements, it's unclear which element to include in the record. As a result, the record in the output payload will be empty.
Array	Record	Array	A record from the input payload that is defined as an array in the schema file will be included in the output payload and stored as an array.
Array	Array	Array	An array in the input payload defined as an array in schema file is treated as an array in the output payload.

Examples of data type rules for Avro conversion when the schema defines a field as a record.

The examples in this section are based on the following schema:

```
{
 "type": "record",
 "name": "ContactAndKids",
 "fields": [
 {
 "name": "MyContact",
 "siebelName": "Contact",
 "type": {
 "type": "record",
 "name": "Contact",
 "fields": [
 {
 "name": "ContactId",
 "type": "string"
 },
 {
 "name": "First_Name",
 "type": "string",
 "siebelName": "First Name"
 }
 {
 "name": "Account",
 "type": {
 "type": "record",
 "name": "Account",
 "fields": [
 {
 "name": "Name",
 "type": "string"
 }
 ]
 }
 }

 ]
 }
 }
 ]
```

```
}
```

Input Payload	Output Payload	Explanation

<pre>{
"Contact": {
"First Name": "Jack",
"Last Name": "Doe",
"ContactId": "1234",
"Account": [
{
"Name": "Test Account1",
"CSN": "88-36MIAD"
}
]
}
}</pre>
	<pre>{
	"MyContact": {
	"ContactId": "1234",
	"First_Name": "Jack",
	"Account": {
	"Name": "Test Account1"
	}
	}
	}</pre>
		The example schema defines both **Contact** and **Account** as records. However, in the input payload **Contact** is defined as a record and **Account** is defined as a single element array. As a result, **Account** is included as a record in the output payload.

<pre>{
"Contact": {
"First Name": "Jack",
"Last Name": "Doe",
"ContactId": "1234",
"Account": [
{
"Name": "Test Account1",
"CSN": "88-36MIAD"
},
{
"Name": "Test Account2",
"CSN": "88-36MIAP"
}
]
}
}</pre>
	<pre>{
	"MyContact": {
	"ContactId": "1234",
	"First_Name": "Jack",
	"Account": {
	"Name": ""
	}
	}
	}</pre>
		In this sample input payload, **Contact** is defined as a record and **Account** is defined as an array with two elements. As a result, **Contact** is included as a record in the output payload, while **Account** is not included.

Examples of data type rules for Avro conversion when the schema defines a field as an array.

The examples in this section are based on the following schema:

```
{
 “type”: “record”,
 “name”: “ContactAndKids”,
 “fields”: [
 {
 “name”: “MyContact”,
 “siebelName”: “Contact”,
 “type”: {
 “type”: “record”,
 “name”: “Contact”,
 “fields”: [
 {
 “name”: “ContactId”,
 “type”: “string”
 },
 {
 “name”: “First_Name”,
 “type”: “string”,
 “siebelName”: “First Name”
 },
 {
```

```
"name": "Account",
"type": {
"type": "array",
"items": {
"type": "record",
"name": "Account",
"fields": [
{
"name": "Name",
"type": "string"
}
]
}
}
}

]
}
}
]
}
```

Input Payload	Output Payload	Explanation

<pre>{
"Contact": {
"First Name": "Jack",
"Last Name": "Doe",
"ContactId": "1234",
"Account": {
"Name": "Test Account1",
"SingleRecordAsJSONObject": "true",
"CSN": "88-36MIAD"
}
}
}</pre>
	<pre>{
	"MyContact": {
	"ContactId": "1234",
	"First_Name": "Jack",
	"Account": [
	{
	"Name": "Test Account1"
	}
	]
	}
	}</pre>
		In this input payload, **Account** is included as a record, even though it is defined as an array in the sample schema. As a result, the output payload contains **Account** as a single element array.
		
		Therefore, if the schema defines a field as an **array**, but the input payload includes it as either an array or a **record**, the output payload will represent the field as an array.

Rules for Duplicate Schema Fields in Payload

This topic addresses scenarios where the payload contains duplicates, and selection is based on the child element.

If the schema fields are duplicated in the payload, the following rules apply for node selection:

- When the child node is defined as a record in the schema.

  Sample schema for examples:

```
{
"type": "record",
"name": "ContactAndKids",
"fields": [
{
"name": "MyContact",
"siebelName": "Contact",
"type": {
"type": "record",
"name": "Contact",
"fields": [
{
"name": "First_Name",
"type": "string",
```

```
"siebelName": "First Name"
},
{
"name": "Account",
"type": {
"type": "record",
"name": "Account",
"fields": [
{
"name": "Name",
"type": "string"
},
{
"name": "ID",
"type": "string",
"siebelName": "CSN"
}

]
}
}

]
}
}
]
}
```

- The duplicates are of type record: A child node that does not have duplicate fields is selected. If no such node exists, blank values are returned in the output. For example:

Input Payload	Output Payload	Explanation

<pre>{
"Quote": {
"Name": "qt1234",
"Contact": {
"First Name": "Jack",
"Account": {
"SingleRecordAsJSONObject":
"true",
"CSN": "88-36MIAD"
}
}
},
"Communication": {
"Name": "qt1234",
"Contact": {
"First Name": "Jill",
"Account": {
"Name": "Test Acc21",
"CSN": "88-36MILE"
}
}
}
}</pre>
	<pre>{
	"MyContact": {
	"First_Name": "Jill",
	"Account": {
	"Name": "Test Acc21",
	"ID": "88-36MILE"
	}
	}
	}</pre>
		In this example, based on the sample schema, **Account** is duplicated. However, **Communication → Contact → Account** includes both fields, **Name** and **CSN,** as configured in the schema, while **Quote → Contact → Account** containsonly the **CSN** field. As a result, the final output contains values from **Communication → Contact → Account**.

- The duplicates include both records and single-element arrays: A child node that does not have duplicate fields is selected. If no such node exists, blank values are returned in the output. For example:

Input Payload	Output Payload	Explanation

<pre>{
"Quote": {
"Name": "qt1234",
"Contact": {
"First Name": "Jack",
"Account": {
"Name": "Test Account2",
"SingleRecordAsJSONObject":
"true",
"CSN": "88-36MIAD"
}
}
},
"Communication": {
"Name": "qt1234",
"Contact": {
"First Name": "Jill",
"Account": [
{
"Name": "Test Acc21",
"CSN": "88-36MILE"
}
]
}
}
}</pre>
	<pre>{
	"MyContact": {
	"First_Name": "",
	"Account": {
	"Name": "",
	"ID": ""
	}
	}
	}</pre>
		In this example, based on the sample schema, **Account** is duplicated. Account appears as a record under **Quote** and as an array under **Communication**. However, because a single array element is treated as a record and both **Account** instances contain all the fields, the final output displays blank values.

- The duplicates include both records and multi-element arrays: A record or a single-element array with at least one schema field is selected. If no such record exists, blank values are returned in the output. For example:

Input Payload	Output Payload	Explanation

<pre>{
"Quote": {
"Name": "qt1234",
"Contact": {
"First Name": "Jack",
"Account": [
{
"Name": "Test Acc1",
"CSN": "88-36MIAD"
},
{
"Name": "Test Acc2",
"CSN": "88-36MILD"
}
]
}
},
"Communication": {
"Name": "qt1234",
"Contact": {
"First Name": "Jill",</pre>
	<pre>{
	"MyContact": {
	"First_Name": "Jill",
	"Account": {
	"Name": "Test Acct21",
	"ID": ""
	}
	}
	}</pre>
		In this example, **Account** under **Quote** is a multi-element array and **Account** under **Communication** is a record. In this case, **Account** under **Communication → Contact** is considered as it has one required schema field, **Name,** in it. As a result, the final output displays the values from **Communication → Contact → Account.**

Input Payload	Output Payload	Explanation

"Account": {
"Name": "Test Acct21"
}
}
}
}		

  - The duplicates only consist of single-element arrays: A child node that does not have duplicate fields is selected. If no such node exists, blank values are returned in the output.
  - The duplicates only consist of multi-element arrays: The output returns blank values.
  - The duplicates include both single-element arrays and multi-element arrays: A child node that does not have duplicate fields is selected. If no such node exists, blank values are returned in the output.

- When the child node is defined as an array in the schema, and:

  Sample schema for examples:

```
{
 "type": "record",
 "name": "ContactAndKids",
 "fields": [
 {
 "name": "MyContact",
 "siebelName": "Contact",
 "type": {
 "type": "record",
 "name": "Contact",
 "fields": [
 {
 "name": "First_Name",
 "type": "string",
 "siebelName": "First Name"
 },
 {
 "name": "Account",
 "type": {
 "type": "array",
 "items": {
 "type": "record",
 "name": "Account",
 "fields": [
 {
 "name": "CSN",
 "type": "string"
 },
 {
 "name": "Name",
 "type": "string"
 }
 ]
 }
 }
 }

 ]
 }
 }
 ]
```

```
}
```

- The duplicates are of type record: A child node that does not have duplicate fields is selected. If no such node exists, blank values are returned in the output.
- The duplicates include both records and single-element arrays: A child node that does not have duplicate fields is selected. If no such node exists, blank values are returned in the output. For example:

Input Payload	Output Payload	Explanation

<pre>{
"Quote": {
"Name": "qt1234",
"Contact": {
"First Name": "Jack",
"Account": [{
"SingleRecordAsJSONObject":
"true",
"CSN": "88-36MIAD"
}]
}
},
"Communication": {
"Name": "qt1234",
"Contact": {
"First Name": "Jill",
"Account": {
"Name": "Test Acc21",
"CSN": "88-36MILE"
}
}
}
}</pre>
	<pre>{
	"MyContact": {
	"First_Name": "Jill",
	"Account": [
	{
	"CSN": "88-36MILE",
	"Name": "Test Acct21"
	}
	]
	}
	}</pre>
		In this example, **Account** under **Quote** is a single-element array and **Account** under **Communication** is a record. As a result, the final output displays the values from **Communication → Contact → Account** as it includes all fields from the schema definition.

- The duplicates include both records and multi-element arrays: A child node that does not have duplicate fields is selected. If no such node exists, blank values are returned in the output. For example:

Input Payload	Output Payload	Explanation

<pre>{
"Quote": {
"Name": "qt1234",
"Contact": {
"First Name": "Jack",
"Account": [
{
"Name": "Test Acc1",
"CSN": "88-36MIAD"
},
{
"Name": "Test Acc2",
"CSN": "88-36MILD"
}
]
}
},
"Communication": {
"Name": "qt1234",
"Contact": {
"First Name": "Jill",</pre>
	<pre>{
	"MyContact": {
	"First_Name": "Jack",
	"Account": [
	{
	"CSN": "88-36MIAD",
	"Name": "Test Acc1"
	},
	{
	"CSN": "88-36MILD",
	"Name": "Test Acc2"
	}
	]
	}
	}</pre>
		In this example, **Account** under **Quote** is a multi-element array and **Account** under **Communication** is a record. The final output displays the values from the multi-element array as it includes all fields from the schema definition.

Input Payload	Output Payload	Explanation

"Account": {
"Name": "Test Acc21"
}
}
}
}		

- The duplicates consist only of multi-element arrays: A child node that does not have duplicate fields is selected. If no such node exists, the blank values are returned in the output. For example:

Input Payload	Output Payload	Explanation

{
"Quote": {
"Name": "qt1234",
"Contact": {
"First Name": "Jack",
"Account": [
{
"Name": "Test Acc1",
"CSN": "88-36MIAD"
},
{
"Name": "Test Acc2",
"CSN": "88-36MILD"
}
]
}
},
"Communication": {
"Name": "qt1234",
"Contact": {
"First Name": "Jill",
"Account": [
{
"Name": "Test Acc11",
"CSN": "88-36MITD"
},
{
"Name": "Test Acc21",
"CSN": "88-36MILL"
}
]
}
}
}
}
	{
	"MyContact": {
	"First_Name": "",
	"Account": [
	{
	"CSN": "",
	"Name": ""
	}
	]
	}
	}
		In this example, both **Quote** and **Communication** have the child node **Account,** which includes the fields **Name** and **CSN**. As a result, none of these fields are considered and the output contains only blank values.

- The duplicates consist only of single-element arrays: A child node that does not have duplicate fields is selected. If no such node exists, blank values are returned in the output.
- The duplicates include both single-element arrays and multi-element arrays: A record or a single-element array with at least one schema field is selected. If no such node exists, blank entries are returned in the output.

Recommended Operational Practices

Following are few recommended operational practices:

- While editing aieventconfig.txt file, it is suggested that one validates it with JSON lint or from Notepad++ or any other online tool to ensure it is syntactically correct and that it does not have unwarranted special characters introduced.
- Set the correct JAVA_HOME for Kafka server to start properly.
- In case of failures, one can check EventLogger file in AI sidecar and the appropriate component (OM) log file of Siebel server to know what went wrong. By default, only error logs are dumped. However, the user may increase the logging level as follows:
  - Sidecar AI: Update the log4j2-siebel.xml file and set the logging level to INFO for EventLogger.
  - OM: Increase the logging level for the OM component you are using to 5.
- aieventconfig file OM passwords: Check the OM mode first and then use the encrypted password for the security authentication profile type (DB or ldap) of the OM.
- When a new AI sidecar is created, server.xml file should have all different ports other than the base AI, otherwise the new AI sidecar will not start properly.
- Start sidecar AI after Kafka server is running and stable.
- When configuration changes are done, services need to be started:
  - aieventconfig.txt file changes > restart only corresponding sidecar AI
  - eventpayloadconfig.txt file changes > restart the specific Siebel Server
- In Web Tools while creating Business Service, copy Event Handler and modify such that Business Service Name, Display name, Display Name String Override all be same.
- Topic names in aieventconfig file and in the Event Subscription view's Event field value should match.
- In Linux OS , make sure to give appropriate permission for below:
  - eventconfig folder and eventpayloadconfig.txt file
  - aiconnection folder and aiconnection.txt file
  - Kafka server (broker) configuration changes (number of partitions, for example) : restart sidecar AIs

Troubleshooting and Debugging

This topic describes various error messages, their potential reasons and troubleshooting tips to solve them.

Issues Observed	Reasons	Troubleshooting Tips
Errors seen in EventLogger during 2-step: `"com.siebel.swsm.eventhandler.AIKafkaConsumer:`	No OM alias for topic configured for 2-step in aieventconfig.txt file.	Add/update entry for OM alias in aieventconfig.txt for the topic configured for 2-step.

Issues Observed	Reasons	Troubleshooting Tips

`callUtilFunction OM not found”`
 	
`“com.siebel.swsm.eventhandler.AIKafkaConsumer:handleNonSiebelConsumerRecord Failed to call executeEvent function com.siebel.swsm.eventhandler.AIKafkaConsumer:lambda$run$0 Failure handling Kafka consumer record”`		

Event messages don’t arrive in correct Kafka topic.	Mismatches between topic entries in aieventconfig.txt and eventpayloadconfig.txt	EventLogger log file will show entry of event messages arriving in one topic but not in the other. Also, consumption of data into consumers will not show in EventLogger. Check for possible mismatches between topic entries in aieventconfig.txt and eventpayloadconfig.txt

Event messages don’t arrive in Kafka and the following logs in EventLogger at start up:
 	
`“com.siebel.swsm.eventhandler.AIKafkaConsumer:handleNonSie Failed to call executeEvent function com.siebel.swsm.eventhandler.AIKafkaConsumer:lambda$run $0 Failure handling Kafka consumer record”`
 	
or
 	
`"Password field is empty. Please update the value in aieventconfig.txt"`
	Wrong password for default OM alias in aieventconfig.txt	Add/update password for default OM alias in aieventconfig.txt

Final event message not seen/received in Kafka topic.	• Topic is not configured in aieventconfig.txt for 1- Step. • Topic mentioned in Event Subscription screen not configured in aieventconfig.txt for 2-Step.	Increase log level to INFO and restart sidecar AI. During start-up of AI sidecar, you may see logs like: “Topic received from OM not subscribed to: <topic name>” in INFO mode. Update aieventconfig.txt file with missing details

Following errors in the appropriate Siebel Object Manager log:
 	
`TLS handshake to SAI failed <hostname>:<port>”`
 	
or
 	
`"Connection refused”`
	Incorrect AI sidecar information is configured in aiconnection.txt for Siebel server.	Enter/update appropriate sidecar info in aiconnection.txt

On Linux, environment the variables are not seen coming into effect though set in **.cshrc file**. Following logs can be found:	Login script not executed	Execute shell login script (for example, run **source .cshrc**).

Issues Observed	Reasons	Troubleshooting Tips

• In EventLogger file:
 	
`[INFO ] <timestamp> [Thread-3] EventLogger - com.siebel.swsm.eventhandler.AIServer:extractSubscriberDat Total number of events subscribed to, received from OM: 0`
 	
`[INFO ] <timestamp> [Thread-3] EventLogger - com.siebel.swsm.eventhandler.AIServer:extractSubscriberDat No topic received from OM`
 	
• In Object Manager component log:
 	
`GenericLog GenericInfo 3 0000008363b6ec5c:0 <date> <time> CSSEAIEventHandler::DoInvokeMethod SIEBEL_EVENT_PUBSUB is not set`		

Error in the EventLoggerfile: `[ERROR] EventLogger - com.siebel.swsm.connmgr.EventSessionStore:getSession You have exceeded the total number of event pools configured: 3 Please check event pool size in aieventconfig.txt file for topic: <topic_name>`	Session pool size is the maximum number of JDB sessions that can be created concurrently while consuming an event on that topic. This error means that the creation of a new JDB session failed as the number of sessions that can be created for the topic is exhausted; that is, the number of active sessions for the topic is equal to the value configured in the `SessionPoolSize` parameter for the topic.	Increase the `SessionPoolSize` for that topic in aieventconfig.txt

Error Message in EventLogger and Debugging Tips

The following table entails error messages that may appear in EventLogger file logging errors reported by AI sidecars.

Error Message in EventLogger	Details and Debugging Tips
Error while getting data that Kafka producer sent.	Sidecar AI received the request from the OM but encountered an error while parsing it. Please check entries in the eventpayloadconfig.txt file. Increase logging level if needed and then check for the following log in EventLogger: "Data received" or "Sending data to Kafka broker from the producer queue". This should print the data received in the request.
Exception during Kafka write	Check the exception message present along with this error message. As a starting point, please check if the Kafka broker is up and running.
Kafka consumer read error	Kafka consumer received the data from the broker but encountered an error while reading it. The error message will also report the exception raised.
Failure handling Kafka consumer record	Kafka consumer received the data from the broker but encountered an error while reading it.

Error Message in EventLogger	Details and Debugging Tips
Failed to handle consumer record	Kafka consumer received the data from the broker but encountered an error while reading it. The error message will also have the exception seen.
Failed to execute event function/Kafka event execute failed/Failed to call execute event function	Kafka consumer in AI sidecar received data from Kafka broker but encountered an error while passing the request to the Siebel server. The error message may also list the exception encountered. Please check the appropriate Object Manager logs if required.
No Siebel comp found. Invalid AI config file	In aieventconfig.txt file, check if correct Siebel components (For example, SCCObjMgr_enu or EAIObjMgr_enu) are present for all topics.
Failure handling Kafka consumer record	Kafka consumer received the data from the broker but encountered an error while reading it.
Failed to handle consumer record	Kafka consumer received the data from the broker but encountered an error while reading it. The error message will also have the exception seen.
Failed to execute event function/Kafka event execute failed/Failed to call execute event function	Kafka consumer in AI sidecar received data from Kafka broker but encountered an error while passing the request to the Siebel server. The error message may also list the exception encountered. Please check the appropriate Object Manager logs if required.
No Siebel comp found. Invalid AI config file	In aieventconfig.txt file, check if correct Siebel components (For example, SCCObjMgr_enu or EAIObjMgr_enu) are present for all topics.
Exception while decrypting Kafka password	In the applicationinterface.properties file, you have to use the encrypted password. If needed, please regenerate the Kafka password once more using EncryptString.jar library and try again. Additionally, check the Java versions being used. Check the exception that is printed along with this error message.
Error in reading AI event config data from file	An IO Exception occurred while reading the aieventconfig.txt file. Please check the file including its R/W access and other settings. Check the exception that is printed along with this error message.
Exception while reading event topic	Please see the exception printed along with this error message. In general, check the "Topic" section in the aieventconfig.txt file.
Siebel component is empty. Invalid AI config file	In aieventconfig.txt file, Siebel component has missing fields which need to be updated. For example, the alias or the user/password may be missing.
Environment variable SIEBEL_EVENT_PUBSUB_PAYLOAD_STRUCT is not set. Defaulting to 1 (old payload)	INFO category error, notifying that the environment variable SIEBEL_EVENT_PUBSUB_PAYLOAD_STRUCT is not set.
Environment variable SIEBEL_EVENT_PUBSUB_PAYLOAD_STRUCT is Empty. Defaulting to 1 (old payload)	INFO category error, notifying that the environment variable SIEBEL_EVENT_PUBSUB_PAYLOAD_STRUCT is set to an empty string.

Error Message in EventLogger	Details and Debugging Tips
Invalid value $$ for Environment variable SIEBEL_EVENT_PUBSUB_PAYLOAD_STRUCT. Defaulting to 1 (old payload)	INFO category error, notifying that the environment variable SIEBEL_EVENT_PUBSUB_PAYLOAD_STRUCT is set to an invalid value other than the valid values of 1 or 2.
SIEBEL_EVENT_PUBSUB_PAYLOAD_STRUCT is set to 2	INFO category error, notifying that the environment variable SIEBEL_EVENT_PUBSUB_PAYLOAD_STRUCT is set to 2.
SIEBEL_EVENT_PUBSUB_PAYLOAD_STRUCT is set to 1	INFO category error, notifying that the environment variable SIEBEL_EVENT_PUBSUB_PAYLOAD_STRUCT is set to 1.

13 Siebel CRM and Coherence Enterprise Caching

Siebel CRM and Coherence Enterprise Caching

This chapter includes the following topics:

- *Introduction to Coherence Cache*
- *Overview of the Coherence Cache in Siebel CRM*
- *About Coherence Cache in Siebel CRM*
- *SSH Tunneling*
- *Overview of Steps to Set Up Coherence with Siebel CRM*
- *Installing and Configuring the Coherence Server*
- *Coherence Server Health Check*
- *Setting Up Siebel Coherence Cache Client*
- *Enabling the Feature on the Siebel Server*
- *Using the Cache with eScripts*
- *Troubleshooting*
- *Restrictions*

Introduction to Coherence Cache

Coherence is a clustered, high-availability, in-memory caching solution from Oracle. It stores in memory data that's accessed often using unique keys, reducing the need for repeated database queries. Coherence follows a client-server architecture, where the server manages the cached data, and the client interacts with it.

In the Siebel architecture, the Coherence **server** handles the actual caching, while the **coherence**—integrated within Siebel—communicates with the server to fetch or update data. This tight integration ensures faster access to commonly used data, improving overall performance.

Coherence offers various cache types, including **Distributed Cache**, **Near Cache**, **Local Cache**, and **View Cache**, each with unique advantages**.** These are configured through XML files. Coherence functionality is fully driven by these XMLs, making it highly flexible and extensible. Any required behavior can be introduced or customized by updating the XML configuration files without writing a single line of code.

Siebel supports coherence Caching from 25.6 onward and primarily uses **Distributed Cache**, where cached data is partitioned and distributed across multiple nodes. These nodes reside on separate physical or virtual machines. As Coherence is an in-memory caching solution, the maximum amount of data the cache can store depends on the number of nodes configured. Each node's capacity is an XML configurable parameter.

Running a single Coherence node enhances performance by minimizing data access latency. Running multiple nodes adds fault tolerance and high availability. These multiple nodes form a Coherence cluster that seamlessly communicates to provide a unified caching experience to the client, even when distributed across different machines. If resources (CPU and RAM) allow, one Coherence node can also run on the same machine as the Siebel server. Because Coherence is an in-memory caching solution, the maximum amount of data the cache can store depends on the number of nodes configured. Each node's capacity is a configurable parameter.

By default, Coherence provides data redundancy by backing up cache data from one node to another. This ensures cache availability even in the event of a node failure. To fully use Coherence's capabilities, **Siebel recommends deploying at least three Coherence nodes.**

The Coherence Server operates based on two key XML configuration files that control its behavior and functionality.

1. **custom-override.xml**
2. **custom-cache-config.xml**

Similarly, the Siebel's Coherence Client relies on the following XML file to access Coherence services.

1. coherence-extend-config.xml

These XML files are Created and managed via the **Siebel Management Console (SMC)** using the **Coherence Cache Server Profile** and **Cache Client Configuration Profile.**

To find out more about Coherence see *Oracle Coherence* on the Oracle website.

To find, see Oracle Coherence Documentation on Oracle Help Center.

Overview of the Coherence Cache in Siebel CRM

Siebel CRM caches some of its metadata in the Coherence cache. The cache is enterprise-wide (all Siebel components use it) but can be enabled for distinct components if needed. The Siebel CRM stores and retrieves values held in the cache rather than from the database. This improves the performance of the application in the following areas.

- Logging into the application through the UI or our APIs
- Components start-up
- Navigation between views
- Transaction throughput through our APIs

We've optimized the code behind application logins. These optimizations allow faster logins, whether through the UI or through our APIs such as REST. This helps with the throughput using our APIs because we've remediated the disproportionate time spent logging into the application when requests were made.

When a Siebel component loads, it requests information from the database to perform its duties. Retrieving these values from the database is slower than getting that same information from the Coherence cache. The more data that's needed for a component, the faster it will start when it gets those values from the cache. When the cache first starts, this information is in the database only, but as the first component requests them, they're loaded into the cache so most of the performance price is paid with the first request for metadata. All requests that follow use the cache. Therefore, when the next component starts, it's much faster. Some cached data is specific to a user context making their use of the application swifter. We cache:

- User Responsibilities
- User Positions

- User Organizations
- User Preferences
- S_RR_* object definitions

Navigating between views means the application must load the compiled object definitions for all the Views, Applets, Controls, Business Components, Links, Tables, et cetera, to construct the View. Because we're caching those definitions, navigation becomes faster.

> **Note:** Siebel has a feature that automatically loads new configuration when it's migrated to the Runtime Repository. The default loop looks every 15 minutes (this interval is configurable) to see if there's a new version of the MAIN Workspace and if so, loads those object definitions to the next navigation. This still holds true with caching and hasn't changed. The new definition is loaded into the cache and replaces the previous definition so that all users, upon their next navigation or sign in, see the updated configuration.

About Coherence Cache in Siebel CRM

Coherence is a caching solution that hosts and serves string values to the Siebel CRM. When a component starts and requests metadata such as Applet, Business Component, Business Service et cetera compiled definitions, it looks to see if the data is in the cache already. If the data is not in the cache, Siebel CRM queries its database and puts the values in the cache. From then on, the values will be retrieved more quickly from the cache. As each value is stored in the cache, operations become faster.

Similarly, when a user logs into the web application, the Object Manager requests the user context along with the underlying business logic such as Applets, Views, Business Components, and other definitions to render the GUI and data. If this information is not already in the cache, Siebel CRM queries the database and stores the retrieved values in the cache. As a result, subsequent navigations become significantly faster.

The cache is any of the nodes you configured to hold your cached data. Oracle recommends at least three nodes for availability and redundancy. Even though there are multiple nodes, we refer to them collectively as “the cache”. The Siebel server connects to the cache by finding what machines are designated in Coherence Cache Client profiles deployed to the Siebel enterprise using the Siebel Management Console. Communication between the Siebel Server and the cache is done using SSH tunneling to keep the transactions encrypted and secure. For SSH tunnelling to work, SSH software is installed on the Siebel Server in client mode, and on each Coherence node machine in server mode.

SSH Tunneling

Communication between Siebel and the Coherence Server isn't secure by default. Coherence provides TLS support; however, it's not available for C++ clients such as Siebel. Therefore, we use SSH tunneling to secure the communication channel between Siebel and the Coherence Server. This requires a SSH server to be running on the Coherence Server host and SSH client on the Siebel host. The SSH client is installed on Windows 10 and later versions by default, so you don't need to install it again. The SSH server must be explicitly installed on the Coherence Server host on Windows. However, you can deploy the SSH server with just a single click in the installer package. See **To install SSH on each Coherence node** in the chapter *Installing and Configuring the Coherence Server*.

On Linux, the SSH client and server are typically already installed, so there's no need for a separate SSH installation. SSH tunneling supported fully from OpenSSH version 7.x and later. Most modern Linux distributions include supported versions by default. Siebel recommends using Oracle Linux Server 8, which comes with OpenSSH_8.0p1.

Overview of Steps to Set Up Coherence with Siebel CRM

This is the sequence to set up Coherence to work with Siebel CRM. Each step will be expanded.

1. Set up Coherence Nodes
    a. Install the Siebel on each Coherence node so that the Siebel Management Console is available.
    b. Install the SSH software on each Coherence node in Windows. Linux usually has this software installed.
    c. Configure and deploy an Enterprise Cache Server Profile on each Coherence node.
2. Set up and Deploy Enterprise Cache Client Profile
3. Enable the feature on the Siebel Server.
4. Restart the Siebel Server.

Note:

- Deploy a Coherence Cache *Client* Profile on the Siebel Server machine using the Siebel Management Console.
- Deploy Coherence Cache *Server* Profiles on each Coherence node using the Siebel Management Console installed on each node.

Set Up Coherence Nodes

Coherence nodes must be set up before configuring and deploying Siebel for Coherence integration. Your first step is to define the machines that will host the Coherence nodes. If the Siebel application is configured to use Coherence but Coherence is not properly set up, errors will occur due to the cache server not being found.

Note: If Siebel fails to connect to Coherence under any circumstances, it automatically starts using the Database.

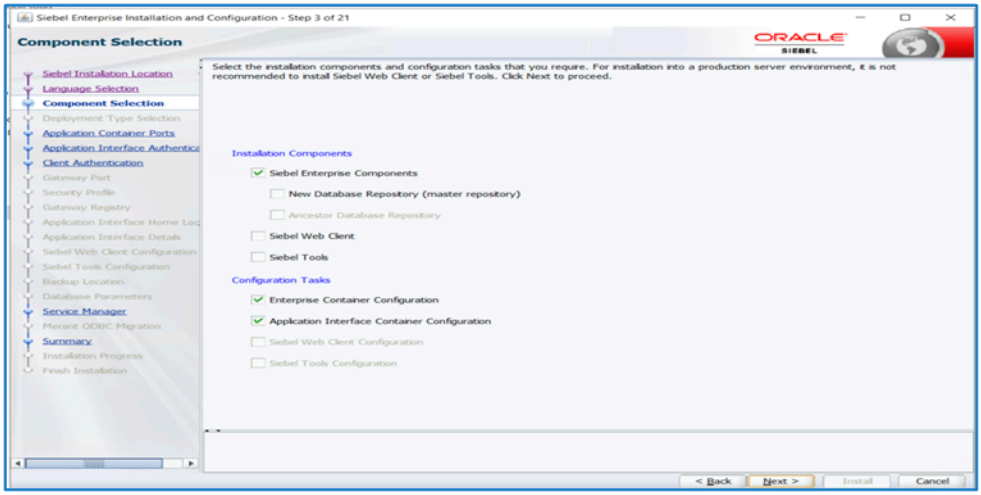
Installing and Configuring the Coherence Server

The Coherence Server is a standalone Java application. To simplify deployment, management, and configuration, we've integrated it with the Siebel Application Internal. Coherence runs as a container within the Application Internal. As a result, installing Coherence requires installing the Siebel Application Internal. The steps to install Coherence are the same as those for installing the Siebel Application Internal.

Install Coherence on Each Node

1. Run the Siebel Installer on the Coherence node.
2. When the installer gets to the Component Selection screen, choose the following three components only.
    a. **Installation Components** section
        i. Siebel Enterprise Components

b. **Configuration Tasks** section
   i. Enterprise Container Configuration
   ii. Application Interface Container Configuration



3. Click **Next** and set the Fields as you did in the installer when you installed the Siebel Server. This is the same Siebel installer and will use the same values as your Siebel installation.
4. Exit when the installation is complete.

Install SSH on Each Coherence Node

1. Once the Coherence installation completes, go to `<Siebel Install Directory>\ses\siebsrvr\BIN`.
2. Find the SSH software installer OpenSSH-Win64-v9.8.1.0.msi and simply double-click it. There will be no installation screens as this is a silent install.
3. The installer will install and start the SSH Server.
4. The SSH Server runs as a standard Windows Service.
5. To verify, press Win + R, type services.msc, and press Enter to open the Services Management Console.
6. Ensure the following services are running:
   a. OpenSSH SSH Server – the actual SSH server daemon (sshd)
   b. OpenSSH Authentication Agent – manages SSH private keys

Linux typically has SSH software installed and running by default.

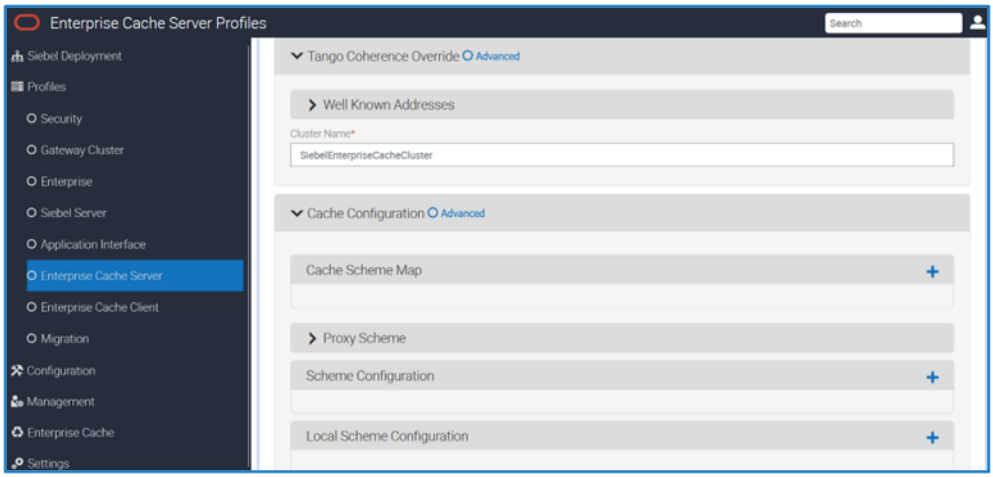
Note: Do this for all your Coherence nodes in Windows. Oracle recommends at least three nodes.

Create the Enterprise Cache Server Profile

You'll create and deploy on each Coherence node an Enterprise Cache Server Profile. The same values must be applied to every Coherence node. When the profiles are deployed, they'e written to Zookeeper and compiled into an XML file so that the Coherence server uses them as input to start the cache service. You'll see different sections, such as **Tango Coherence Override** and **Cache Configuration**.

To create and deploy an Enterprise Cache server profile on each Coherence node:

1. Sign into the Siebel Management Console (SMC).
2. Go to **Profiles**.
3. Create an Enterprise Cache server profile.
4. Enter the following values and deploy the Enterprise Cache server profile.



Tango Coherence Override > Well Known Addresses

Profile Field	Suggested Value	Description
Server Address	IP address for the node hosting the Coherence	The well-known address helps cluster discovery and enables seamless registration of new nodes into the Coherence cluster.
Cluster Name	siebel_cluster	This can be custom, but it must be consistent across all Coherence Nodes so that they join the same cluster.

Cache Configuration > Cache Scheme Map

Profile Field	Suggested Value	Description
Cache Name	siebelcache	The name for your cache. By default, all components use this name, and all profiles must use the same value. Change it only if you've a good reason to do so.
Scheme Name	distributed	This is the scheme that will define the cache settings.

Cache Configuration > Proxy Scheme

Profile Field	Suggested Value	Description
Service Name	ExtendTcpProxyService	Default: ExtendTcpProxyService – This is the name for the Coherence service that will handle traffic. Don't change this value. The extend proxy service (**`ProxyService`**) is a cluster service that allows Siebel to access a Coherence cluster.

Profile Field	Suggested Value	Description
Maximum Thread Count	75	Default: 75 – This is the amount of Maximum threads Proxy will use. Don't change this value. This setting enables thread pooling for handling proxy operations.
Minimum Thread Count	10	Default: 10 – This is the minimum number of threads proxy will use. Don't change this value. This setting enables thread pooling for handling proxy operations.
Cache Server Host	127.0.0.1	This is the local loopback address where the Coherence server is running for this node. Don't change this value. This must be same for every Coherence node.
Port	7077	The port on which Coherence is running and listening for traffic. You can use any available port.
Auto Start Extended Service	True (checked)	Default: True (checked) – This allows the Coherence service to automatically start. Don't change this value.

Cache Configuration > Scheme Configuration

Profile Field	Suggested Value	Description
Distributed Scheme	distributed	This value defines how data is stored and retrieved from the Cache. Siebel uses Distributed Cache. This must match the value in the Scheme Name field from the Cache Configuration section.
Type	Backup Scheme	Default: Backup Scheme – This is Coherence's way of making sure the data is available either from a disk or (in our case) in memory. Don't change this value.
Service Name	DistributedCache	This is the way Coherence ensures that data will not be blocked from retrieval because of bottlenecks. Don't change this value.
Auto Start Distributed Service	True(checked)	Default: True(checked) – This allows the distributed service to start automatically. Don't change this value.
Backup Scheme Configuration	Local	The name of the scheme that will be defined in the next section. This specifies the name of another scheme to inherit from.

Cache Configuration > Local Scheme Configuration

Profile Field	Suggested Value	Description
Local Scheme Name	**Local**	This value defines the scheme for the backup scheme defined earlier. This value must match the one defined earlier.
Eviction Policy	Hybrid	Default value: Hybrid – combines both Least Recently Used (LRU) and Least Frequently Used (LFU) principles to find out which cache entries to evict when the cache is full.
Maximum Cache Size (in MB)	2000	Default value: 2000 – The amount of memory the cache will use before starting to evict data. This is per Cache.

Maximum Cache Size and Guidance on Sizing of Coherence Node Machines

When Application Internal Tomcat is used to deploy the Coherence Server, it is essential to configure both the minimum and maximum heap size settings, as the cache data is stored within the heap space of the Application Internal process.

To determine the appropriate minimum and maximum heap size, first calculate the maximum cache size using the following formula. The parts of this calculation are elaborated on below.

Max Cache Size = Max Number of Entries × (Average Cache Size per Entry + 150 bytes)

Where:

- **Max Cache size** is the capacity of data held in memory as part of the caching mechanism.
- **Number of Entries** = Total Metadata + Customized Metadata + (Cache Entries per User Login × Total Users Base)
- **Average Cache Size per Entry** = Average Cache size of Siebel key-value pairs
- **150 bytes** = Additional overhead to account for estimation inaccuracies.

In Siebel, two types of data are cached: metadata and user contexts. These Cache entries are created during components start up, when users log in and when they navigate to views or access business logic for the first time after Coherence Caching is enabled.

Component startup and view navigation create cache entries related to metadata sourced from the S_RR tables. Each record in the S_RR* tables generate one key–value pair in the cache upon access, and each key-value pair represents a single cache entry.

For metadata, the maximum number of cache entries corresponds to the total number of records from all the S_RR tables, along with any additional metadata that may be created in the future—either by modifying existing metadata records or by creating new ones.

For example, suppose a customer is on Version MAIN 20 and upgrades to Build 25.6 or later, with Coherence Caching enabled. When navigating to *My Accounts*, if there have been no changes on SIS Account List Applet from Main Version 0 through Version 20, the metadata cache entry is created as:

MAIN-WSID | SIS Account List Applet | Applet | 0 | ENU

If a migration occurs and the version increments to MAIN 21, and further customizations are made to **My Accounts** in that version, a new cache entry is created upon first access. This entry remains active until it is updated again.

MAIN-WSID | SIS Account List Applet | Applet | 21 | ENU

This behavior applies to all metadata components that undergo further changes.

Therefore, when calculating cache sizing, you should account for the probability of additional entries due to customization:

- **Highly customized environments:** ~50% of total metadata
- **Moderately customized environments:** ~25% of total metadata
- **Lightly customized environments:** ~5–10% of total metadata

These percentages can be used as an initial estimate for determining the number of cache entries for sizing calculations.

The following example illustrates how to calculate the **number of cache entries.** An internal PSR environment is used as a reference. The total metadata entries amount to **1,052,526 ,** calculated by retrieving the record count from all **S_RR*** tables and summing the results. We used the below provided SQL statement, and you can follow the same approach.

```
SELECT table_name,
 num_rows AS approx_row_count
FROM user_tables
WHERE table_name LIKE 'S\_RR%' ESCAPE '\'

UNION ALL

SELECT 'TOTAL' AS table_name,
 SUM(num_rows) AS approx_row_count
FROM user_tables
WHERE table_name LIKE 'S\_RR%' ESCAPE '\'

ORDER BY table_name;
```

Let's assume a medium customization environment, approximately **25% of 1,052,526**, which is **263,131 entries**, should be considered as additional metadata. Similarly, during user login, **7 cache entries** are generated: 1 for user context, 2 for profiles, 3 for preferences, and 1 for responsibilities. Using these counts, let's calculate the **maximum number of entries** that could be created in the cache for a 25k User Base.

1. **Number of Cache Entries:**

   Total Cache Entries=Total Metadata + Customized Metadata + (Cache Entries per User Login × Number of Users)

   Substituting the values: `=1,052,526+263,131+(7×25,000)= 1,490,657`
2. **Max Cache Size:**

   Max Cache Size=Total Cache Entries × (Average Entry Size + 150 bytes)

   From our analysis, the average cache size per entry is 5021 bytes.

   Substituting the values:

   Max Cache Size= `1,490,657 × (5021 + 150) bytes=7,708,187,347 bytes ≈ 7.2GB`

   Set the calculated Max Cache Size in **SMC** > **Enterprise Cache Server Profile** > **Max Cache Size** field.

For a 3-node Coherence setup, backup is enabled by default. Therefore, the total Max Cache Size becomes 14.4 GB (7.2 GB x 2), resulting in approximately 4.8 GB per node for a medium-load environment.

3. **Heap Size:**

   Using the Max Cache Size value, the required heap size is calculated as follows:

   Approximately **30%** of the 1GB JVM's memory is typically allocated as **working heap**, with an additional **10%** reserved to accommodate potential estimation inaccuracies. Therefore, while calculating the required heap size, you should add an estimated 40% overhead to the maximum Cache Size to ensure stable operation without memory pressure.

   Total Heap Size `=7.2 GB + (40% × 7.2 GB)=10.08 GB ≈ 10.1 GB`

   If the Coherence server is running on a single node, set the heap space to 10.1 GB. For a 3-node Coherence cluster, the maximum cache size is doubled due to the default backup configuration. Therefore, the maximum cache size is (7.2 GB x 2) or 14.4 GB. The total heap size is calculated as 40% of 14.4 GB, which is then added to the original 14.4 GB. This value is then divided by the 3 nodes in the cluster.

   ```
   14.4GB + (40% × 14.4GB)=20.16GB

   20.16 GB/3=6.7 GB per node
   ```

   Set this value for the Application Internal process. Refer to the *Performance Tuning Guide* → *Tuning Apache Tomcat* section for detailed steps to configure the heap space settings.

   The above formula applies to a single-language environment. In a multilingual environment, the above calculated heap space will not be sufficient for the same use case because cache entries are generated separately for each language.

   Only language-independent entries are shared across all languages, whereas language-dependent metadata and user context entries are generated per language in the Cache.

   Therefore, the formula to calculate the total number of cache entries in a multi-lingual environment is discussed in the following sections.

4. **Total Cache Entries for a Multilingual Siebel Environment**

   Total Cache Entries = `(LI Metadata + (LD Metadata × No_of_Languages)) + (%Customized_Metadata of Total_Data) + [(LI UserEntries + (LD UserEntries × No_of_Languages)) × No_of_Users]`

   Here:

   LI=Language Independent

   LD=Language Dependent

   No_of_Languages = Total Number of Languages

   The below example illustrates the Max Cache Size calculation for a 3-language Siebel environment using Oracle's internal reference implementation. The analysis shows 34,884 Language-Independent (LI) entries and 1,017,642 Language-Dependent (LD) entries. Since the environment supports 3 languages, the LD entries

are multiplied by 3 to account for language-specific metadata that get stored in the cache. Additionally, the percentage of customized metadata is assumed to be 25% of the total metadata.

For User Context, there are 3 cache entries per user that are language-dependent. Therefore, the language dependent User Context entries must also be multiplied by the number of supported languages (3). Here, the user base is considered 25K.

Total Cache Entries = `(34884 + (1017642 × 3)) + 263,131) + [(4 + (3 × 3)) × 25000]`

`=3,350,941+ 325,000`

`= 3,675,941`

5. **Max Cache Size for Multilingual Siebel Dump**

   Max Cache Size= `3,675,941×5121`

   `=18,824,493,861 = 17.5 GB`

   Hence Heap Space Required is `17.5 GB + 40% of 17.5 GB =24.5 GB`

   If the Coherence server is running on a single node, set the heap space to **24.5 GB**. For a 3-node Coherence cluster, the maximum cache size is doubled due to the default backup configuration. Therefore, the maximum cache size is (17.5 GB x 2) or 35 GB. The total heap size is calculated as 40% of 35 GB, which is then added to the original 35 GB. This value is then divided by the 3 nodes in the cluster:

   `35 GB+(40%×35 GB)=49 GB`

   `49 GB/3=16.3 GB per node`

   The above Max Cache Size and Heap Space calculations are suitable for most customer environments, as they represent an average estimate. In some customer environments, the user base may be smaller with a higher percentage of customization and users navigating through most of the Siebel Views. Conversely, in other environments, the user base may be larger about one hundred thousand with minimal customization, where users primarily access only business-critical views. These factors generally balance each other out.

   For an accurate calculation of the maximum cache size, determine the total metadata record count from your Siebel database, estimate the expected percentage of future metadata customizations, and identify the total number of users. Substitute these values into the cache size formula. The average cache size per entry can be set to 5021 bytes.

Advanced Settings

Some customers will have already installed Coherence for other reasons and might've advanced knowledge of the settings they require for their Coherence cache. In these cases, we've provided an **Advanced** option. When you click this option, you don't have to fill out the individual fields. You might already have a Tango override file to use. In the fields here such as “Tango Configuration XML” and “Cache Configuration XML”, you can simply paste your already created XML. This is a simple way to do this.

Tip: Once you've created and deployed a single Enterprise Cache Server profile, the SMC application writes the entered settings to XML files. The **Tango Configuration XML** is stored in the `custom-override.xml file`, and the **Cache Configuration XML** is stored in `custom-cache-config.xml file`. Both files found at:<SIEBEL Installation directory>\SES\applicationcontainer_internal\webapps\cgcohconf. You can use the contents of this file to paste into your next Coherence node’s Advanced option.

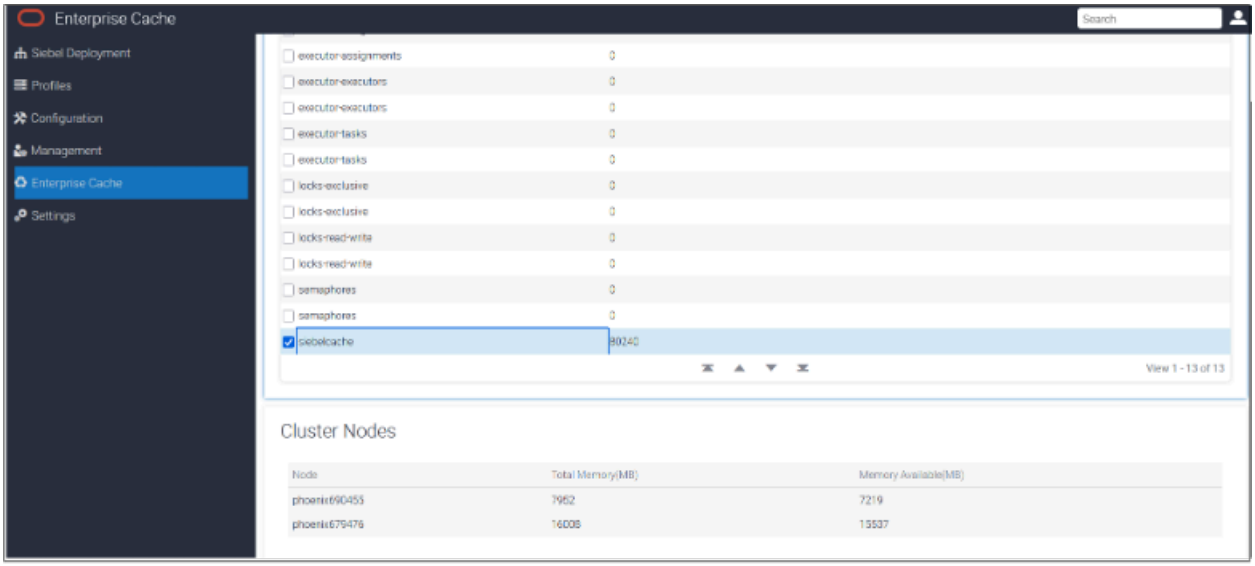
Deploy the Enterprise Cache Server Profile to your enterprise

To deploy the Coherence Server, In SMC, select **Siebel Deployment**, then choose **Enterprise Cache Server**. Fill in the following fields and submit the form.

Deployment Field	Suggested Value	Description
Host Name: Gateway Port	CoherenceServerName:Gateway port	This is the Coherence Server machine name and Gateway port where the Coherence server will be deployed. You can find the gateway port in the **`gateway.properties`** file for the Value Gateway at **`<Siebel Installation Directory>\ses\applicationcontainer_internal\webapps`**.
Profile	Choose your Profile	N/A
Action	Deploy	Starts the Coherence Server.
Enterprise Cache Server Agent Node Name	Any Name*	Any name that's self descriptive.
Enterprise Cache Server Agent Description	Any Value*	Any meaningful description.

Coherence Server Health Check

After deploying the Coherence Server, you can verify that it is up and running by checking its status in SMC under the Enterprise Cache tab. This page displays the total number of cache entries, the number of nodes on the cluster, the total memory allocated to each Coherence Node, and the available memory in megabytes . If a Coherence node is down, its entry will be missing from this page. This node information is dynamic and reflects the current state. To view the latest information, refresh the SMC Web Page.



You can also verify if it's up and running by interacting with the cache using the Coherence Console.

Run the following command to start the Coherence console. Once inside, use the command `cache <cacheName>` to connect to the cache. If successful, the cache configuration details will be displayed. The `help` command provides details about available commands for interacting with the Coherence Server, such as `get <key>, put <key>`, and more.

Windows:

```
java.exe -server -showversion -Dcoherence.pof.enabled=true -
Dcoherence.distributed.localstorage=false -Dcoherence.management=all -
Dcoherence.management.remote=true -
Dtangosol.coherence.cacheconfig=<SIEBEL_ROOT>\ses\applicationcontainer_internal\webapps\cgcohconf\coherence-
cache-config.xml -
Dtangosol.coherence.override=<SIEBEL_ROOT>\ses\applicationcontainer_internal\webapps\cgcohconf\custom-
override.xml -
cp <SIEBEL_ROOT>\ses\applicationcontainer_internal\webapps\siebel\WEB-INF\lib\coherence-25.03.2.jar com.
tangosol.net.CacheFactory
```

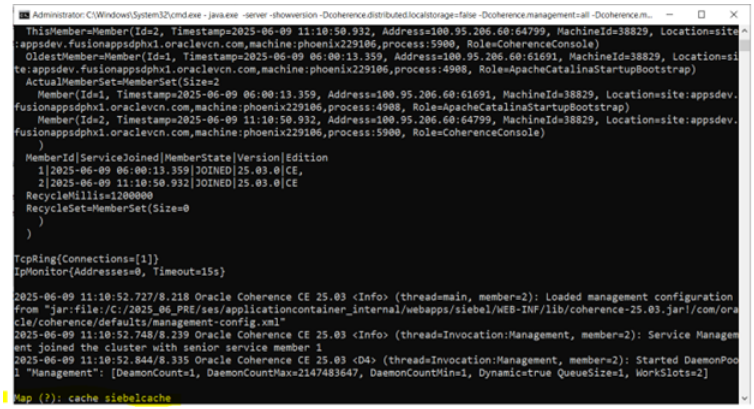
Linux:

```
./java -server -showversion -Dcoherence.pof.enabled=true -
Dcoherence.distributed.localstorage=false -Dcoherence.management=all -
Dcoherence.management.remote=true -Dtangosol.coherence.cacheconfig=<SIEBEL_ROOT>/ses/
applicationcontainer_internal/webapps/cgcohconf/coherence-cache-config.xml -
Dtangosol.coherence.override=<SIEBEL_ROOT>/ses/applicationcontainer_internal/webapps/cgcohconf/custom-
override.xml -
cp /<SIEBEL_ROOT>/ses/applicationcontainer_internal/webapps/siebel/WEB-INF/lib/coherence-25.03.2.jar
 com.tangosol.net.CacheFactory
```

After running the command, you'll see `Map(?)` as shown below. Simply type the command shown in the image: `cache <cachename>`.

`Example: cache siebelcache.`

Similarly, type help to view more available commands.



This command is supported on both Windows and Linux Coherence Server deployments.

> **Note:** Coherence recommends using Java version 17 or higher. JAVA_HOME should be: `<Siebel Installation Directory>\ses\jre\21.0\`

Setting Up Siebel Coherence Cache Client

You need to configure the Coherence client in Siebel to connect to and use the Coherence cache service.

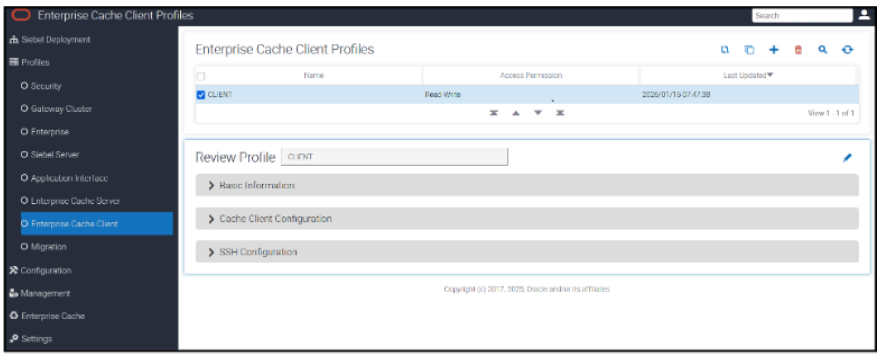
This involves creating the appropriate cache client configuration files, defining the cache name, and ensuring that the client can discover and communicate with the Coherence cluster.

Creating the Coherence Cache Client Profile

Just as you created profiles for each of your Coherence nodes, you must create a client profile for your Siebel server which will be a client of the cache. This profile instructs the Siebel server how to find and use the Coherence cache.

The profile consists of 3 sections:

- Basic Information
- Cache Client Configuration
- SSH Configuration



Basic Information

This defines the logging level for the Enterprise Cache Client Profile and its deployment actions. Logs are written to the CacheClient file under **\ses\applicationcontainer_internal\logs**. The supported log levels are ERROR, INFO, and DEBUG.

Cache Client Configuration

The Cache Client Configuration Section contains details of the cache instance Name and the Proxy service Name and Coherence nodes on which it is running. *For additional information about the proxy service, please refer to the topic "Proxy Services in Coherence Servers." below.*

The values entered in Cache Client Configuration Section are stored in ZooKeeper and are written to the `coherence-extend-config.xml` file under `\ses\siebsrvr\ADMIN` - during Enterprise Cache Client deployment. This file is then read by Siebel to establish a connection with Coherence. Siebel reads this file during initialization; therefore, if this section is modified, a Siebel service restart is mandatory for the changes to take effect.

The Siebel application connects to the Coherence service using the Coherence Extended Proxy Client which comes with the Siebel build starting with version 25.6, which in turn connects to the Proxy Service running on the Coherence Server. The Coherence Server hosts only two services the Extended **Proxy Service** and the **Cache Service**. The Proxy Service routes client requests to the Cache Service running in the cluster for operations such as get, put, and remove.

The Extended Proxy Client running in the Siebel Server uses the Enterprise Coherence Client Profile to locate the proxy services, cache name, and the addresses/ports of the Coherence Server nodes in the cluster to access and perform cache operations. Hence, the Enterprise Coherence Client Profile specifies a service name, cache name, Addresses and Port.

Out of the box, the proxy service name is ExtendTcpProxyService and the cache name is siebelcache.

The following section explains proxy services in more detail and describes how they guide the proxy client in establishing the connection.

Proxy Services in Coherence Servers

A Proxy Service in Coherence is a Server-side service that allows Extended Client to access a Coherence cluster. An Extended Client cannot join the Coherence cluster directly; instead, it connects to a Proxy Service, which communicates with the other services within the cluster.

At any point in time, the proxy client maintains a connection to only one proxy service, through which all requests are routed. If this proxy service becomes unavailable because of a network interruption or related failure, the proxy client automatically fails over and connects to another proxy service hosted on a different Coherence node. Hence the address and port of the additional Proxy Service must be specified in the TCP Configuration section of the Enterprise Cache Client Profile. For this reason, Siebel recommends configuring all 3 Coherence node addresses and ports in the TCP Configuration section to ensure high availability in a 3-node cluster.

By default, in Siebel, every Coherence server is launched with a Proxy Service. This can be seen in the Proxy Scheme section of the Enterprise Cache Server Profile.

The default address used in the TCP configuration is "localhost" as the connection is secured via SSH. This is required because we employ the SSH Local Port Forwarding mechanism. With this mechanism, Siebel running in the local host first connects to the Port specified in the TCP Configuration section; the traffic from this Port is forwarded by SSH to the corresponding port on the remote Coherence Server through an SSH tunnel.

Refer to *https://www.ssh.com/academy/ssh/tunneling-example#local-forwarding*, to understand SSH Local port forwarding.

SSH Configuration

The SSH Configuration section contains details such as the application internal host name and the port of the Coherence server , Local SSH Port, SSH Remote Port, SSH Server Port and SSH Username . Using this information, SSH tunnels are created between the Siebel Server and each Coherence node.

SSH Server Configuration must be completed on all three Coherence nodes. Each Coherence node should have its own TCP and SSH Configuration entry in the Enterprise Cache Client Profile so that, if the first proxy service is unavailable, the Extended Client can seamlessly switch to another proxy service running on a difference coherence Node through the appropriate SSH tunnel.

SSH uses public key–based authentication. In this mechanism, SSH authenticates a specific user account on the remote Coherence machine. Without an SSH Username, the SSH authentication fails; therefore, the SSH Username must be specified.

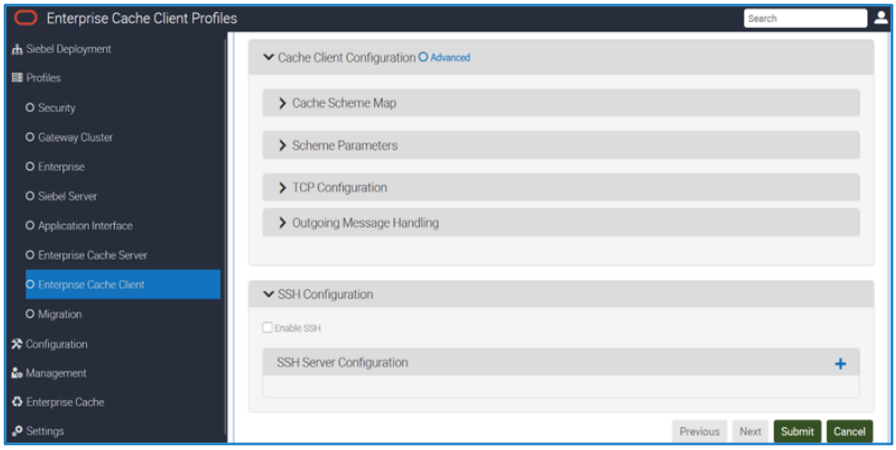
The SSH tunnel works independently of Siebel; therefore, redeploying the profile is enough to make changes in this section effective.

Note: The terms “Coherence server” and “Coherence Node” are used interchangeably throughout this chapter.

Refer *https://www.ssh.com/academy/ssh* details on the SSH tunnel.

To Create and Deploy an Enterprise Cache Client Profile

1. Sign in to the SMC for your Siebel Server.
2. Create a new Enterprise Cache Client profile.
3. Enter the following values:



Cache Client Configuration > Cache Scheme Map

Profile Field	Suggested Value	Description
Cache Name	siebelcache	This value must match the value used in the Cache Name field in your Enterprise Cache Server Profiles. By default, All the Siebel component is configured to use the cache name “siebelcache”. Siebel recommends same for both Cache Server Profile and Cache Client Profile.

Profile Field	Suggested Value	Description
Scheme Name	remote	This makes it clear that this is intended to point to a remote machine. Whatever value you pick, it must be re-used when we define this scheme in Scheme Parameters section

Cache Client Configuration > Scheme Parameters

Profile Field	Suggested Value	Description
Scheme Name	remote	This makes it clear that this is intended to point to a remote machine. This must match the value chosen in the previous section.
Service Name	ExtendTcpProxyService	Default: ClientExtendTcpProxyService – This is the service that Siebel CRM will use to communicate to Coherence.

Cache Client Configuration > TCP Configuration

Profile Field	Suggested Value	Description
Cache Server Host	**127.0.0.1**	This represents the loopback address used for the SSH tunnel—which is always 127.0.0.1. Don't change this Value for any nodes
Port	**7077**	This is the port number on which Siebel listens via the SSH tunnel. Siebel recommends using a unique port for each node when creating SSH connections—typically in the 707x range; for example 7077 for Node 1, 7078 for Node 2, and so on.

Cache Client Configuration > Outgoing Message Handling

Profile Field	Suggested Value	Description
Heartbeat Interval	30s	Default: 30s - Specifies the interval between ping requests. A ping request is used to ensure the integrity of a connection. **Don't change this value.**
Heartbeat Timeout	15s	Default: 15s - Specifies the maximum amount of time to wait for a response to a ping request before declaring the underlying connection unusable. **Don't change this value.**
Request Timeout	30s	Default: 30s - Specifies the maximum amount of time to wait for a response message before declaring the underlying connection unusable. **Don't change this value.**

Profile Field	Suggested Value	Description
Enable SSH	Check this checkbox	Enables SSH tunneling.

Profile Field	Suggested Value	Description
Cache Server Host Name Port	Coherence Node:GatewayPort	Specify the name and Gateway port of one of your Coherence nodes. You need to do this for each node. This tells Siebel CRM how to find the Coherence cache machines.
SSH Local Port	The port which this Siebel server is using for SSH traffic	This is the port number on which Siebel listens via the SSH tunnel. This should be as same as Port in the TCP Configuration. You can use any available port for the connection. Siebel recommends using ports in the 707x range—for example, 7077 for Node 1, 7078 for Node 2, and so on.
SSH Remote Port	The port which this Coherence Server is using for SSH traffic	When you set up your Coherence nodes you specified a port that the nodes use. Enter that port value here.
SSH Server Port	Default is 22. It can be changed to any available port on the Coherence server.	This port is used by the SSH client on the Siebel server to connect to the SSH server running on the Coherence server.
SSH UserName	Machine Login User	Specify a user who is a Machine administrator and has access to both the Siebel machines and the Coherence server machines. The SSH username is a global setting, applicable to all SSH Server Configuration entries. Therefore, the same username is used for all tunnels **NOTE**: Ensure that this user has logged in at least once to all Siebel servers in the enterprise and the Coherence servers before Enterprise Cache Client Profile Deployment. This is mandatory because the private keys are generated during Deployment under the user's profile directory (C:\Users\<username>), and SSH tunnels are created in the context of that user. Otherwise, the SSH tunnel will not be established.

From the Siebel server, you must create one SSH tunnel per Coherence node. Each tunnel needs its own unique local port on the Siebel server (e.g., 7070, 7071, 7072). Therefore, the SSH Local Port must be different in each SSH configuration created for a Coherence node to establish its tunnel.

If you reuse the same SSH local port for more than one tunnel, ports can collide and Siebel cannot reliably route traffic to different Coherence nodes. For example, if the first-connected node fails, Siebel cannot reach the remaining Coherence nodes at startup, resulting in a complete cache outage. The port is how Siebel distinguishes different Coherence Nodes and selects the correct tunnel to reach right destination when using SSH local port forwarding.

> **Note:** The SSH Remote Port may be the same (or different depending on your setup) in each SSH configuration for every Coherence node since they run on different machines. The only requirement for SSH Remote Port is that, it must match the Port defined in the Proxy Scheme for that Coherence server in Enterprise Cache Server Profile.

If you have three Coherence nodes, for example, create three SSH configuration sections in the SMC, each with a different SSH Local Port for example, 7070, 7071, and 7072.

Example mapping with different Siebel Server local SSH ports mapping to three Coherence nodes where they have each used the same SSH port on their separate machines.

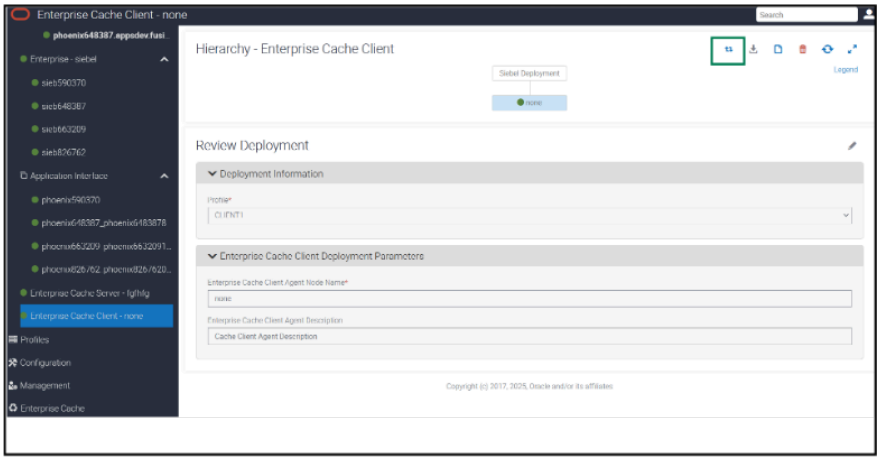
Tunnel #	Local (Siebel Server) Port	Remote SSH Port (Coherence)Node
1	localhost:7070	CoherenceNode1:7077
2	localhost:7071	CoherenceNode2:7077
3	localhost:7072	CoherenceNode2:7077

From version 26.3 onwards, you can deploy the Coherence Client Profile from the Siebel Deployment Tab. This provides the advantage of creating and updating Coherence Client Profile and starting SSH tunnels from all Siebel servers within the Enterprise in a single Deployment.

A Recreate Tunnel button allows SSH tunnels to be recreated across all Siebel servers in the enterprise for an already deployed profile, without impacting the Siebel Enterprise ecosystem. This enables quick re-establishment of tunnels in case of network outages or related issues with a single click.

Additionally, this also eliminates the need to restart individual Siebel servers across the Enterprise to recreate SSH tunnels which was mandatory until Siebel version 26.2 when Coherence Caching enabled.

If you are already using Siebel with Coherence caching and have upgraded to version 26.3, it is mandatory to create and deploy a new Enterprise Cache client profile. Otherwise, the tunnel will not be created. This will not impact the existing Siebel Coherence ecosystem; it will only recreate the tunnel.



To Deploy Enterprise Cache Client Profile to Enterprise

To deploy the Cache Client Profile in SMC, select **Siebel Deployment** > **Enterprise Cache Client**. Fill in the following fields and submit the form.

Deployment Field	Value	Description
Profile	Choose your Profile	N/A

Deployment Field	Value	Description
Action	Deploy	Creates /Updates coherence-extend-config.xml Creates SSH Tunnels
Enterprise Cache Client Agent Node Name*	Any Name*	Self-Descriptive Any Name
Enterprise Cache Client Agent Description	Any Value*	Meaningful any Description

Enterprise Cache Client Deployment triggers the following actions:

1. Create or update the cache client file `coherence-extend-config.xml` under `<SIEBEL_ROOT>\ses\siebsrvr\ADMIN` on all Siebel servers in the enterprise.
2. Generates key files using the Elliptic Curve Digital Signature Algorithm (ECDSA) and creates the public key file `id_ecdsa.pub` on all Siebel machines in the enterprise under **C:\Users\<username>\.ssh** on Windows. On Linux, the corresponding is **/home/<username>/.ssh.**
3. Copies the public key file and creates the `administrators_authorized_keys` file in Coherence Servers.
4. Appends the public key contents from `id_ecdsa.pub` to **C:\ProgramData\ssh \administrators_authorized_keys** in Coherence Server in Windows. Similarly in Linux, **/home/User/.ssh/ authorized_keys**.
5. When a custom SSH server port (for example, 44) is configured instead of the default port 22 in the Enterprise Cache Client Profile, the `C:\ProgramData\ssh\sshd_config` file is automatically updated and the OpenSSH service is restarted to apply the change on Windows. On Linux, this update must be carried out manually. Refer to the *SSH Tunnel in Linux* section below for details.
6. SSH tunnel is established from each Siebel Server to the Coherence Servers.
7. Separate Tunnel is created to every Coherence node configured in the SSH Configuration Section.

   Once the tunnel is created, you can verify this by checking the **Details** tab in Task Manager, where you'll see `ssh.exe` running on Windows. Similarly, on Linux, you can verify the tunnel by running `ps-ef|grep ssh`.
8. The SSH tunnel looks like the following in Task Manager Command line tab in Windows and Linux:

   Windows:

   ```
   <SIEBEL_ROOT>\ses\siebsrvr\bin\ssh.exe -i C:\User\<UserId>\.ssh\id_ecdsa -C -N -c aes128
   -ctr -L 127.0.0.1:<SSH Local Port>:127.0.0.1:<SSH Remote Port >
   <UserId>@<CoherenceHostName> -p <SSH Server Port> -o StrictHostKeyChecking=no
   ```

   Linux:

   ```
   ssh -i /export/home/User/.ssh/id_ecdsa -C -N -c aes128-ctr -L 127.0.0.1: :<SSH Local
   Port>:127.0.0.1:<SSH Remote Port <SSH Username>@ CoherenceHostName> -p <SSH
   Server Port> -o StrictHostKeyChecking=no
   ```

SSH Tunnel in Linux

A slight variation is observed in the SSH operational style on Linux.

1. In Linux, the SSH Username must match the OS user account used to start the Coherence Server's Application Internal. Otherwise, the SSH tunnel will not be established.

   For example, if the Coherence Server application Internal is running under the user *`sblqa1`* , the SSH Username must also be `sblqa1` in Enterprise Cache Client Profile.
2. When using a custom SSH server port other than the default port 22, the user must update the Port <custom_port> entry in the /etc/ssh/sshd_config file and reload the SSH service using `sudo systemctl reload sshd`. The `sudo` privileges are required to perform this action.

Note:

a. When a new Siebel Server joins the enterprise, simply redeploy the Enterprise Cache Client.
b. When a new Coherence server joins the cluster, update the details in the Enterprise Cache Client Profile both the TCP Configuration and SSH Configuration sections and then redeploy the profile.

Advanced Options in Coherence Client Profile

Rather than filling out every field individually in this profile in the SMC, you can configure the Coherence client if you have expertise with Coherence, simply define the XML and copy-paste it in **Advanced Options**. This gives you a fast, reliable way to define your Enterprise Cache Client Profiles.

Advanced Options sections where you can paste the XML contents of the file to quickly set up another profile for different Siebel Enterprise using the same values.

Note: Use a consistent method when creating and updating the Enterprise Cache Client Profile. If the profile was created using the field-based option, it should also be updated using the field-based option. Similarly, if the profile was created using the Advanced option, updates should be made using the Advanced option.

Enabling the Feature on the Siebel Server

This feature is disabled by default because at least three machines must be configured to support the Coherence cache, which might take some time. There are several steps to enable this feature when you're ready. You can enable the feature at the enterprise level by setting a hidden enterprise level parameter in Server Manager or the at the UI level.

To enable the caching feature at the enterprise level:

1. Sign in to the Server Manager on the Siebel Server machine.
2. Run the following command:

   change entparam EnableOMDistributedCache=True

Enterprise Parameter Name	Suggested Value
EnableOMDistributedCache	**True** - enables the feature across your Siebel enterprise. **False** or no parameter defined - disables the feature for your Siebel enterprise.

3. Restart the Siebel server.

Note: Siebel's connection to Coherence can be verified from the application UI under **Help** > **Technical Support** > **Cache Status**. The status is displayed as **Connected** when Siebel successfully connects to Coherence, **Not Connected** when the connection fails, and **Not Enabled** when Coherence caching is not enabled. This information reflects the system's current runtime status.

Additionally, when the Siebel service or components start successfully, the initial Object Manager (OM) log shows a successful connection to the Coherence cache named `<CacheName>`, as shown below.

The following log snippet shows a successful Coherence connection.

```
GenericLog GenericError 1 00000002695e1804:0 2026-01-07 04:34:57 Connected to
Coherence Cache with siebelcache name
```

Using the Cache with eScripts

The following two eScript methods (enabled internally) allow developers to manually store and retrieve string values from the server script:

- GetDCache
- PutDCache

For more details, see the Objects Interfaces Reference guide on Oracle Help Center.

Troubleshooting

These are some suggestions to decide if the Coherence cache is operating as expected with Siebel CRM.

Situation	Advice
Find out if the Siebel server is connected to the Coherence cache	In the application log file for the application object manager, you'll see the following message that indicates success connecting to the remote Coherence cache. `GenericLog GenericError Connected to Coherence Cache with <cachename> name`

Find out if the connection from an object manager to the cache failed.	In the application log file for the object manager in question you'll see entries like this:
	
	`UCacheLog UCacheErrorLog [UCACHE] Error encountered in creating the cache 'siebelcache'- Could not establish a connection with the server. Possible cause - Server is not running or POF is not configured at the server side or trying to connect to different port`
	
	`GenericLog GenericError Coherence Cache does not exist with <cachename> name`

Find out if the cache became unreachable after successfully connecting.	If the cache goes down after the Siebel server has successfully connected to it, you'll see entries in the application log file like this: `UCacheLog UCacheErrorLog [UCACHE] Error encountered in inserting key 'ProfAttrState1:0-1'- Could not connect to the server. Please contact Siebel Administrator for troubleshooting.`

For troubleshooting any issues that occur during Enterprise Cache Client profile Submission or Deployment, increase the log level in the Enterprise Cache Client Profile and retry. The CacheClient log captures most of the Enterprise Cache Client–related traces.

The CacheClient log records Enterprise Cache Client activities such as successful generation of the Cache Client configuration file (`coherence-extend-config.xml`), SSH key generation, key exchange with the Coherence server, and SSH tunneling details. Since tunneling is established between the Siebel and Coherence servers, a few events are also logged on the Coherence Server side. Therefore, review both the **CacheClient logs on the Siebel machines** and the **CacheServer logs on the Coherence server machines** for end-to-end analysis.

When a custom SSH server port is configured, the `sshd_config` file is updated with the custom port and the OpenSSH service is restarted. These steps are automatically performed during the Enterprise Cache Client profile deployment on the Coherence server machine on Windows . Therefore, these activities are recorded in the CacheServer log file.

Once the SSH tunnel is established, it runs independently, and we no longer have control over the process and its logs. The only way to verify that the tunnel is functioning properly is to check the Application UI under **Help** > **Technical Support** > **Cache Status**. If the status shows **Connected**, the tunnel is healthy.

If the tunnel goes down for any reason, Siebel will not be able to connect to the Coherence Server. In this case, the Application UI under **Help** > **Technical Support** > **Cache Status** will display **Not Connected**, and below similar errors will appear in the logs. This typically indicates that either the SSH tunnel is not functioning, or the Coherence cluster is down.

Errors in the AOM log when the tunnel or Coherence cluster is down:

```
UCacheLog UCacheErrorLog 1 00000002683d20e8:0 2025-06-01 22:40:06 7884: [UCACHE]
Error encountered in fetching key '|Runtime Metadata Version Info|System Activity Object|0' -
Key does not exist.
```

When encountering this issue, first check the tunnel status on each Siebel Server in the enterprise. If tunnels are down, click the **Recreate Tunnel** button in the already deployed Enterprise Cache Client Profile in the SMC to re-establish the connection. If the tunnels are healthy but the Cache Status still shows **Not Connected**, then verify the Coherence cluster status in **SMC** > **Enterprise Cache** tab.

For more detailed logs, update the `log4j2-siebel.properties` file located at **ses\applicationcontainer_internal \webapps\siebel\WEB-INF** by changing `logger.CacheClient.level = ERROR` to `TRACE`.

After making this change, restart the Application Internal service, and retry the operation. Detailed logs will be available in the configagent and cloudgateway files under `\ses\applicationcontainer_internal\logs`.

Restrictions

Once Coherence has been installed in association with Siebel CRM, you can't use the Coherence cache for custom reasons. You can't, for example, create other applications to use the Coherence cache. The only way you can use Coherence for your requirements is through an eScript. Siebel CRM will use this cache for its own needs and no other application can use the cache.