Siebel

Deployment Planning Guide

August 2021

Siebel
Deployment Planning Guide

August 2021

Part Number: F12700-06

Copyright © 2021, Oracle and/or its affiliates.

Authors: Siebel Information Development Team

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display in any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software" or “commercial computer software documentation” pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government’s use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

The business names used in this documentation are fictitious, and are not intended to identify any real companies currently or previously in existence.

Contents

Preface i

1 What's New in This Release 1

What's New in Siebel Deployment Planning Guide, Siebel CRM 21.8 Update 1
What's New in Siebel Deployment Planning Guide, Siebel CRM 20.12 Update 1
What's New in Siebel Deployment Planning Guide, Siebel CRM 20.8 Update 1
What's New in Siebel Deployment Planning Guide, Siebel CRM 20.6 Update 2
What's New in Siebel Deployment Planning Guide, Siebel CRM 20.1 Update 2
What's New in Siebel Deployment Planning Guide, Siebel CRM 19.1 Update 2

2 Siebel Architecture Overview 3

Siebel Architecture Overview 3
Building Blocks of a Siebel CRM Deployment 3
About Siebel Client Types 6
About the Siebel Application Interface 8
About the Siebel Enterprise Server and the Siebel Server 9
About the Siebel Gateway 11
About the Siebel File System 12
About Siebel Server Load Balancing 13
About Siebel Internet Session Network API 14
User Request Types 14
About the Siebel Connection Broker 15
About the Server Request Broker 17
About the Server Request Processor 18
About Siebel Enterprise Application Integration 18
About Siebel Enterprise Integration Manager 19
About Siebel Tools and Siebel Web Tools 19
Example of User Request Flow in a Siebel CRM Deployment 20
About Siebel Open UI 21

3 Siebel Infrastructure Planning 23

Siebel Infrastructure Planning 23

Process of Infrastructure Planning 23
Determining How Siebel CRM Applications Are Used 23
Defining Data Flows and Integration Requirements 24
Determining Database Requirements 26
Mapping Business Requirements to Siebel Server Components 27
Defining High Availability Policies 27
Mapping Siebel CRM Deployment Elements to Platforms 29
Determining Network Requirements 34
Defining a Test and Transition Plan for the Siebel CRM Deployment 35

4 High Availability Deployment Planning 37
High Availability Deployment Planning 37
How Service Failures Affect the Siebel CRM Deployment 37
About High Availability Deployment Options 43
Recommended High Availability Techniques for Specific Services 45
Recommendations for High Availability Deployments 47
About Resilient Processing 49

5 Server Clustering Planning 51
Server Clustering Planning 51
About Server Clustering 51
Where to Use Server Clustering 53
Recommendations for Server Clustering 53

6 Data Integrity and Capacity Planning 55
Data Integrity and Capacity Planning 55
Sizing the Database for a Siebel CRM Deployment 55
Database Table Planning 56
Database Recovery Planning 58
Database Physical Device Planning 59
Database RAID Array Planning 59

7 Application-Level Deployment Planning 61
Application-Level Deployment Planning 61
Session Communications Server Components 61
Session Communications Performance Factors 62

Session Communications Deployment Planning 63
Siebel Email Response Server Components 64
Siebel Email Response Performance Factors 64
Siebel Email Response Deployment Planning 65
Siebel Product Configurator Server Components 65
Siebel Product Configurator Architecture 67
Siebel Product Configurator Performance Factors 68
Siebel Product Configurator Deployment Planning 69
Siebel Workflow Deployment Planning 77
Planning Batch Processing When Using Siebel Remote 78

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 *http://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 Deployment Planning Guide, Siebel CRM 21.8 Update

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

Topic	Description
Recommendations for Collocating Server Components	Modified topic. Updated information about Siebel CRM server installation and deployment. Also provided more details about application containers and how they might affect collocation decisions for Siebel Server and Siebel Gateway.

What's New in Siebel Deployment Planning Guide, Siebel CRM 20.12 Update

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

Topic	Description
Siebel Product Configurator Architecture	Modified topic. Siebel Constraint Engine (also known as Advanced Constraint Engine) is no longer supported for Siebel Product Configurator, as of Siebel CRM 20.12 Update. Installation and Siebel Management Console configuration options for Siebel Constraint Engine no longer apply. Siebel Product Configurator uses and will continue to use the existing iLog integration. For more information, see *Siebel Installation Guide* .

What's New in Siebel Deployment Planning Guide, Siebel CRM 20.8 Update

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

Topic	Description
Siebel Mobile Web Client	Modified topic. Oracle Database SE2 is now used for the local database, replacing Oracle Database XE. For more information, see *Siebel Installation Guide* .

Topic	Description

What's New in Siebel Deployment Planning Guide, Siebel CRM 20.6 Update

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

Topic	Description
About the Siebel Connection Broker	Modified topic. Updated the descriptions of the LL and RR values for the ConnForwardAlgorithm parameter.

What's New in Siebel Deployment Planning 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.

What's New in Siebel Deployment Planning Guide, Siebel CRM 19.1 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 Architecture Overview

Siebel Architecture Overview

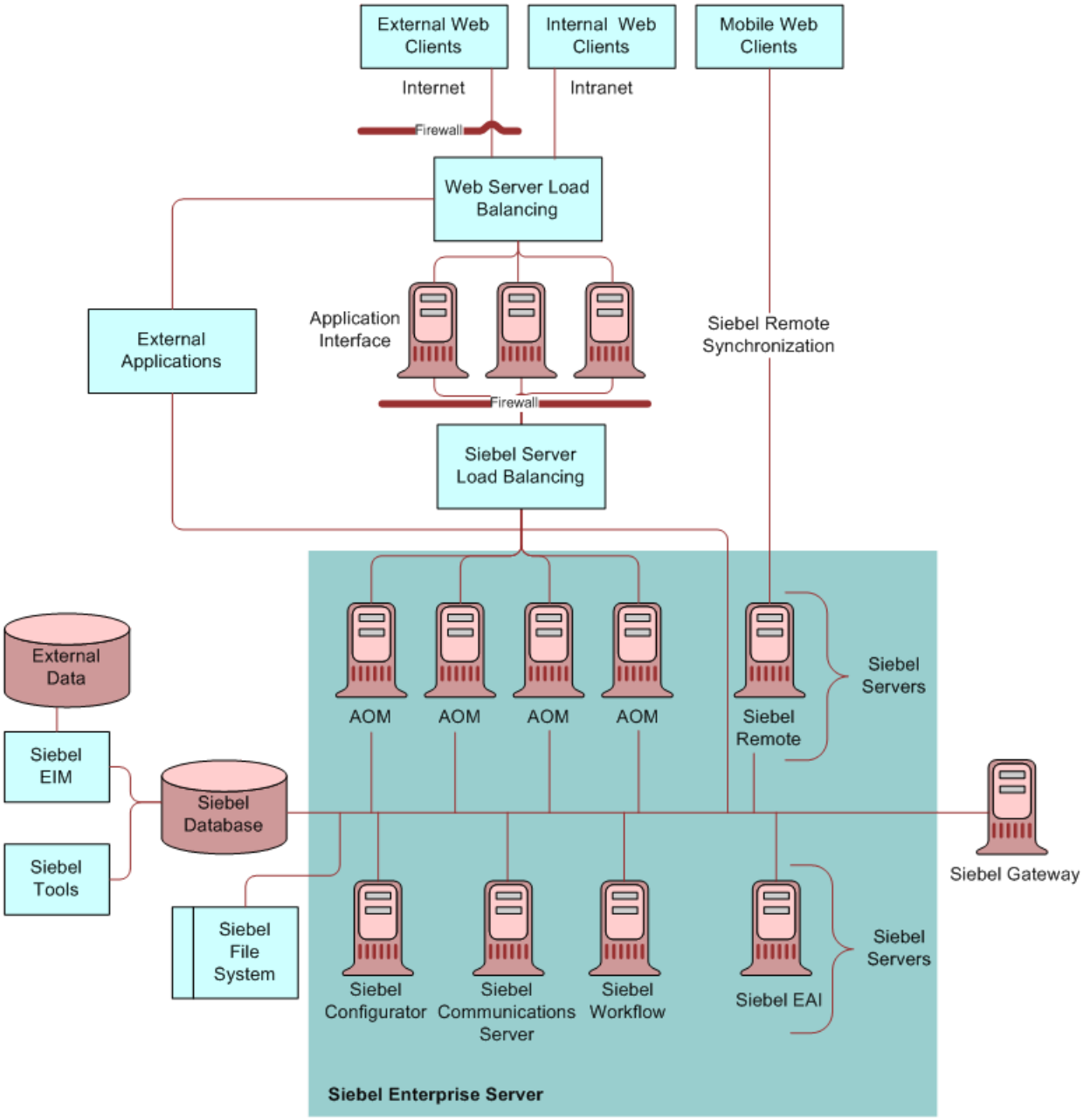
This chapter provides an overview of the architecture for Siebel CRM. It includes the following topics:

- *Building Blocks of a Siebel CRM Deployment*
- *About Siebel Client Types*
- *About the Siebel Application Interface*
- *About the Siebel Enterprise Server and the Siebel Server*
- *About the Siebel Gateway*
- *About the Siebel File System*
- *About Siebel Server Load Balancing*
- *About Siebel Internet Session Network API*
- *About User Request Types*
- *About the Siebel Connection Broker*
- *About the Server Request Broker*
- *About the Server Request Processor*
- *About Siebel Enterprise Application Integration*
- *About Siebel Enterprise Integration Manager*
- *About Siebel Tools and Siebel Web Tools*
- *Example of User Request Flow in a Siebel CRM Deployment*
- *About Siebel Open UI*

Building Blocks of a Siebel CRM Deployment

The following figure shows an example of the elements in a Siebel CRM deployment. A brief description of these elements appears in the following table to help you understand the Siebel CRM architecture.

The current release of Siebel CRM supports certain specific database and operating system platforms, as well as certain combinations of them. For a list of the operating systems and RDBMS products supported by the current release, see the Certifications tab on My Oracle Support.



The following table describes the major elements in a Siebel CRM deployment.

Entity	Description
Siebel Web Clients	Includes the following client types: • Siebel Web Client • Siebel Mobile Web Client • Siebel Developer Web Client (limited support) • Siebel Mobile applications For more information, see *About Siebel Client Types*.

Entity	Description
Siebel Application Interface.	The Siebel Application Interface identifies requests for Siebel data and forwards them to the Siebel Servers. It receives data from Siebel Servers and helps format it into Web pages for Siebel clients. You install and configure the Siebel Application Interface as a separate module, located in the DMZ. (The DMZ or demilitarized zone is located behind an external firewall but outside of an internal firewall.) For more information, see *About the Siebel Application Interface*.
Siebel Server load balancing	Siebel Server load balancing is provided by Siebel Application Interface and Siebel Gateway working together. For more information, see *About Siebel Server Load Balancing*.
Siebel Enterprise Server	A logical grouping of Siebel Servers that connect to one database and that allows management of Siebel Servers as a group. The corresponding configuration entity is called the Siebel Enterprise. For more information, see *About the Siebel Enterprise Server and the Siebel Server*.
Siebel Servers	Application server software that provides both user services and batch mode services to Siebel clients or other components. For more information, see *About the Siebel Enterprise Server and the Siebel Server*.
Siebel Gateway	Stores configuration and status information for Siebel Enterprise, Siebel Server, Siebel Application Interface, and other components. For more information about Siebel Gateway, see *About the Siebel Gateway*. **Note:** Siebel CRM supports an optional native clustering feature for Siebel Gateway to provide high availability benefits to Siebel CRM customers. This feature works at the software level and is the preferred and recommended approach for clustering the Siebel Gateway. For more information, see *Defining High Availability Policies*.
Siebel database	Stores database records. Includes third-party RDBMS software and Siebel tables, indexes, and seed data. Also includes the Siebel runtime repository.
Siebel File System	Shared file system directory or directories storing data and physical files used by Siebel clients and Siebel application components. For more information, see *About the Siebel File System*.
Siebel CRM deployment	All of the physical and logical elements required to deploy Siebel applications, including: • Siebel database • Siebel Gateway • Siebel Enterprise (configuration entity) • Siebel Servers • Siebel Application Interface • Siebel clients and Web browsers • Other components
Siebel Enterprise Integration Management (EIM)	These modules allow importing and exporting of data from other databases to the Siebel database, and provide various other integration services. For more information, see *About Siebel Enterprise Integration Manager* and *About Siebel Enterprise Application Integration*.

Entity	Description
Siebel Enterprise Application Integration (EAI)	
Siebel Tools Siebel Web Tools	Siebel CRM provides two main development tools: • Siebel Tools is a Windows-based, integrated environment for configuring Siebel applications. • Siebel Web Tools is an Application Object Manager component that provides similar functionality through the Siebel Web Client. Most of the information provided about Siebel Tools also applies to Siebel Web Tools. You use Siebel Tools to modify standard Siebel objects and create new objects to meet your organization's business requirements. For example, you use Siebel Tools to extend the data model, modify business logic, and define the user interface. **Note:** For more information about these programs, see *About Siebel Tools and Siebel Web Tools*. See also *Using Siebel Tools* .

About Siebel Client Types

This topic describes the Siebel client types. The term Siebel Web Client sometimes refers to all of the browser-based clients and sometimes refers to one specific client type, which is described in the following section. For information about Siebel Open UI, the user interface for all of the Siebel Web Client types, see *About Siebel Open UI*.

Note: Some client functionality for Siebel Web Client, Siebel Mobile Web Client, and Siebel Developer Web Client might require that other Siebel client software be installed, such as Desktop Integration Siebel Agent (DISA) or Outlook to Siebel Drag and Drop. For information about installing these client modules, see *Desktop Integration Siebel Agent Guide* . See also *Siebel Installation Guide* .

This topic contains the following information:

- *Siebel Web Client*
- *Siebel Mobile Web Client*
- *Siebel Developer Web Client*
- *Siebel Mobile Applications*

Siebel Web Client

The Siebel Web Client runs in a standard browser on the end user's client computer. The browser connects through the Siebel Application Interface to the Siebel Server, which executes business logic and accesses data from the Siebel database. Only the user interface layer of the Siebel CRM architecture resides on the user's computer.

This topic is part of *About Siebel Client Types*.

Other considerations about the Siebel Web Client are as follows:

- **Installed software.** In general, no additional application software is required on the client. At minimum, the client requires only a Web browser.

- **Application connection.** This is the connection through the Siebel Application Interface to the Siebel Server. Applications run on the Siebel Server and forward pages to the client. Applications display in a standard Web browser on the end user's client computer, such as a connected laptop or desktop computer.
- **Database connection.** This is the connection through the Siebel Server to the Siebel database. No Siebel database or database client is installed on the client.

For information about installing and setting up the server environment to support this client, see *Siebel Installation Guide* and other documentation. See also the Certifications tab on My Oracle Support.

Siebel Mobile Web Client

The Siebel Mobile Web Client is like the Siebel Web Client except that it includes installable software and uses a local database that synchronizes with the Siebel database.

This topic is part of *About Siebel Client Types*.

The Siebel Mobile Web Client includes the following:

- **Installed software.** Windows-based software containing Siebel applications and related services is installed on each client. The client also requires a Web browser.
- **Application connection.** Applications run on each client. Applications display in a standard Web browser on the end user's client computer, such as a laptop.
- **Database connection.** A local database and a local Siebel File System are installed on each client. Applications access the local database.

  Users periodically synchronize the local database and Siebel File System with a remote Siebel database and Siebel File System. Users synchronize data using components on the Siebel Server. Mobile users synchronize with the remote Siebel database and the Siebel File System without going through the Siebel Application Interface or any other Siebel Server component. The Siebel Remote (alias Remote) and Disconnected Mobile Synchronization (alias MobileSync) component groups must be enabled on the Siebel Server, among other requirements. The local database for Siebel Mobile Web Client uses Oracle Database SE2.

For information about installing the Siebel Mobile Web Client and the local database and about installing and setting up the server environment to support this client, see *Siebel Installation Guide* . For information about setting up and synchronizing the local database, see *Siebel Remote and Replication Manager Administration Guide* . See also the Certifications tab on My Oracle Support.

Siebel Developer Web Client

The Siebel Developer Web Client is like the Siebel Mobile Web Client except that it connects with the Siebel database for the enterprise.

Note: This client type is supported for development, administration, and troubleshooting purposes only.

This topic is part of *About Siebel Client Types*.

The Siebel Developer Web Client includes the following:

- **Installed software.** Windows-based software containing Siebel applications and related services is installed on each client. The client also requires a Web browser.
- **Application connection.** Applications run on each client. Applications display in a standard Web browser on the end user's client computer, such as a connected laptop or desktop computer.
- **Database connection.** A direct connection to the Siebel database is required. Appropriate database client software must be installed on the client. This client can connect directly to the Siebel File System or connect to it through the File System Manager server component.

For information about installing the Siebel Developer Web Client and installing and setting up the server environment to support this client, see *Siebel Installation Guide* . See also the Certifications tab on My Oracle Support.

Siebel Mobile Applications

The Siebel Mobile applications are a group of Siebel CRM applications that are accessed from a browser on a mobile device, such as a tablet or smart phone, where the browser connects through the Siebel Application Interface to an Application Object Manager component on a Siebel Server.

For Siebel Mobile disconnected applications, the Disconnected Mobile Synchronization component group (alias MobileSync) must be enabled on the Siebel Server, among other requirements.

This topic is part of *About Siebel Client Types*.

For more information about Siebel Mobile applications, see *Siebel Mobile Guide: Connected* and *Siebel Mobile Guide: Disconnected* . For information about setting up the server environment to support these applications, see *Siebel Installation Guide* and other documentation. See also the Certifications tab on My Oracle Support.

About the Siebel Application Interface

The Siebel Application Interface serves as the Web server for Siebel CRM applications. The Siebel Application Interface identifies requests for Siebel application data coming from Web clients and flags these requests for routing to a Siebel Server. When information is sent from the Siebel Server back to the Web client, the Siebel Application Interface helps complete the composition of the Web page for forwarding to the client.

You can install and deploy multiple Siebel languages on a single Siebel Application Interface instance. The Siebel Server and the Siebel Application Interface do not have to be operated in the same language. However, the Siebel Server, the Siebel Application Interface, and all other server components must use the same character set. You install and configure the Application Interface as a separate module, located in the DMZ. (The DMZ or demilitarized zone is located behind an external firewall but outside of an internal firewall.)

The Siebel Application Interface and Siebel Gateway work together to provide Siebel Server load balancing. When a user requests a new application connection, Siebel Application Interface sends a request to Siebel Gateway, which returns a connect string for the least-loaded Application Object Manager from among the Siebel Servers supporting that component. The user session will use this Application Object Manager.

Note: As of Siebel CRM version 17.5, application containers for Siebel Application Interface instances on multiple nodes can be load balanced using Apache HTTP Server (httpd) and Apache Tomcat Connector (mod_jk). For more information, see *Components Involved in Service Failures*.

For information about installing and configuring the Siebel Application Interface and about deploying Siebel languages, see *Siebel Installation Guide* and *Siebel Global Deployment Guide* .

About the Siebel Enterprise Server and the Siebel Server

This topic describes the Siebel Enterprise Server and Siebel Server, and also the Application Object Manager components of the Siebel Server.

For information about applicable installation and configuration tasks, see *Siebel Installation Guide* .

This topic includes the following information:

- *Siebel Enterprise Server*
- *Siebel Server*
- *Application Object Manager*

Siebel Enterprise Server

The Siebel Enterprise Server is a logical grouping of one or more Siebel Servers that connect to one Siebel database. The corresponding configurable entity is called the Siebel Enterprise. You can configure some server parameters at the enterprise level. Such parameters are inherited by individual Siebel Servers and applicable components. Some parameters can be overridden at the server level or component level.

You use the Siebel Enterprise Server installer for installing Siebel Gateway, Siebel Server, Siebel Database Configuration Utilities, Siebel EAI Connectors, Siebel Enterprise Cache, and Siebel Application Interface.

After the initial configuration of the Siebel CRM deployment, using the Siebel Management Console, some subsequent configuration and administration tasks for Siebel Enterprise, Siebel Server, and server components might be performed by one or more administrators using Siebel Server Manager. Server Manager supports both a command-line user interface and a graphical user interface (GUI).

This topic is part of *About the Siebel Enterprise Server and the Siebel Server*.

Siebel Server

Each Siebel Server functions as an application server and is composed of server components. Each server component performs a defined function. Server components or groups of components determine which applications and services a Siebel Server supports. Components run in one of several modes:

- **Interactive mode.** Interactive mode components start tasks automatically in response to user requests. Interactive tasks run until the user ends the session. Examples of interactive components include the Application Object Managers (alias, for example, SCCObjMgr_deu for Siebel Call Center for German) and the Synchronization Manager (alias SynchMgr).

- **Background mode.** Background mode components handle background processing tasks. Typically, background tasks are called by interactive tasks. Background tasks run until they are explicitly shut down. Examples of background components include Transaction Router (alias TxnRoute) and Workflow Monitor Agent (alias WorkMon).
- **Batch mode.** Batch mode components handle processing of asynchronous work requests. When the task is complete, the component exits. Examples of batch components are Database Extract (alias DbXtract) and Enterprise Integration Mgr (alias EIM).

Many of the Siebel Server components can operate on multiple Siebel Servers simultaneously, allowing Siebel applications to scale across many computers to support large numbers of users.

Other Siebel Server components provide additional functionality, including the following:

- Synchronization for Siebel Mobile Web Clients and Siebel Mobile disconnected applications
- Integration with legacy or third-party data
- Automatic assignment of new accounts, opportunities, service requests, and other records
- Workflow management
- Document generation

This topic is part of *About the Siebel Enterprise Server and the Siebel Server*.

Siebel Connection Broker (SCBroker)

The Siebel Connection Broker component provides load balancing of connection requests to multiple Application Object Manager threads or processes running on the same Siebel Server.

Siebel Server Implementation

The Siebel Server runs as a system service under Windows and as a process under UNIX. This system service or process monitors and controls the state of all of the server components on that Siebel Server. Each Siebel Server is one instantiation of the Siebel Server system service or process within the current Siebel Enterprise Server.

Interactive and batch components can be configured to run as multiple processes or in some cases as multithreaded processes. Application Object Manager components (which are interactive) can run as multiple processes and as multiple threads for each process. Background mode components can run as multiple processes only.

For information about administering the Siebel Server system service or process, see *Siebel System Administration Guide* .

Language Pack Installation

It is strongly recommended to install the same set of languages on each server computer in your Siebel Enterprise Server. However, you can deploy different languages on different Siebel Servers, as needed. For more information, see *Siebel Installation Guide* . See also *Siebel Global Deployment Guide* .

Application Object Manager

One of the most important types of server components is the Application Object Manager. These server components always run in interactive mode. They process user requests and are application- or service-specific. For example, the Siebel Call Center component group contains the Call Center Object Manager (alias SCCObjMgr_deu for German, for example), one for each language deployed on the Siebel Server. This Application Object Manager provides the session environment in which this application runs.

Internally, each Application Object Manager also contains a data manager and the Siebel Web Engine. When an Application Object Manager receives a user request to start an application, it follows this procedure:

- The business object layer starts an application user session, processes any required business logic, and sends a data request to the data manager.
- The data manager creates an SQL query and forwards it the Siebel database.
- The data manager receives the data from the database and forwards it to the business object layer for additional processing.
- The business object layer forwards the result to the Siebel Web Engine, which helps create the user interface for the data. The Siebel Web Engine then forwards the Web pages to the Siebel Application Interface.

This topic is part of *About the Siebel Enterprise Server and the Siebel Server*.

Application Object Manager Implementation

An Application Object Manager server component is implemented as a multithreaded process on the Siebel Server. At run time, a parent process starts one or more Application Object Managers as multithreaded processes, according to the Application Object Manager configuration. The terms multithreaded server or MT server are alternative terms for the multithreaded process, which is also called an Application Object Manager process.

Each thread in an Application Object Manager hosts tasks that are typically linked to one user session. These threads might be dedicated to particular user sessions, or they might serve as a pool that can be shared by user sessions. For each Application Object Manager, a few threads are dedicated to housekeeping functions.

Each Application Object Manager task communicates with the Siebel database, the Siebel Application Interface, or other components, as follows:

- Communication with the Siebel database uses ODBC database connections. You can manage and tune database connections for optimal performance. You can optionally configure connection sharing for database connections.
- Communication with the Siebel Application Interface uses SISNAPI (Siebel Internet Session API), a Siebel messaging format that runs on top of the TCP/IP protocol. SISNAPI connections use encryption and authentication based on Transport Layer Security (TLS).
- Communication with other Siebel Enterprise Server components (including other Siebel Servers) also uses SISNAPI.
- The Siebel Connection Broker (SCBroker) on each Siebel Server listens on a static, configurable TCP port for requests coming from the Siebel Application Interface. SCBroker forwards these requests to Application Object Manager processes.

For more information about the operation of multithreaded processes for Application Object Manager components, see *Siebel System Administration Guide* and *Siebel Performance Tuning Guide* .

About the Siebel Gateway

The Siebel Gateway provides the dynamic address registry for Siebel Servers and server components, and also for Siebel Application Interface and other modules. For example, at startup, Siebel Server within the Siebel Enterprise Server stores its network address in the Siebel Gateway's nonpersistent address registry.

Siebel Enterprise Server components query the Siebel Gateway registry for Siebel Server availability and address information. When a Siebel Server shuts down, this information is cleared from the registry.

The Siebel Application Interface and Siebel Gateway work together to provide Siebel Server load balancing. When a user requests a new application connection, Siebel Application Interface sends a request to Siebel Gateway, which returns a connect string for the least-loaded Application Object Manager from among the Siebel Servers supporting that component. The user session will use this Application Object Manager.

The Siebel Gateway also includes persistent storage in the registry for configuration information for Siebel Server, Siebel Application Interface, and other installable components. This information includes:

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

As this configuration information changes, such as during the configuration of Siebel Enterprise, a Siebel Server, or a Siebel Application Interface, this data is written to the Siebel Gateway registry.

There can be only one Siebel Gateway, or one Siebel Gateway cluster, installed and configured for each environment in which you have created a Siebel Enterprise. Further, you cannot share the same Siebel Gateway across development, test, and production environments.

Siebel CRM supports an optional native clustering feature for Siebel Gateway, to provide high availability benefits to Siebel CRM customers. This feature works at the software level and is the preferred and recommended approach for clustering the Siebel Gateway. For more information, see *Defining High Availability Policies*.

For information about installation and configuration tasks associated with Siebel Gateway, see *Siebel Installation Guide* .

It is strongly recommended to install the same set of languages on each server computer in your Siebel Enterprise Server. For more information, see *Siebel Installation Guide* . See also *Siebel Global Deployment Guide* .

Related Topics

Defining High Availability Policies

Related Books

Siebel Installation Guide

Siebel Global Deployment Guide

About the Siebel File System

The Siebel File System is a shared file system directory or set of directories. The Siebel File System stores document files, Siebel Product Configurator models, and other files that are not suitable for database storage. Siebel CRM provides roles and responsibilities to control access to these files. System user preferences are stored in the `userpref` subdirectory of the Siebel File System.

For information about setting up and maintaining the Siebel File System, see *Siebel Installation Guide* and *Siebel System Administration Guide* .

Siebel File System Recommendations

The following list provides recommendations for dealing with the Siebel File System:

- All of the Siebel Servers must have direct access to the Siebel File System. The only exception to this rule is when a server computer is running a Siebel Document Server only. If the File System Manager server component (alias FSMSrvr) is disabled on the Siebel Document Server, then it accesses the Siebel File System through the FSMSrvr on another Siebel Server. In all other cases, the Siebel Server must be able to directly access the Siebel File System.
- The Siebel File System can be configured to use multiple directories that exist on separate devices or partitions. Before you configure the Siebel Enterprise, at least one file system directory must exist that you can designate for use by the Siebel File System.
- There are few strict rules for deploying the Siebel File System. Theoretically, the file system can be hosted on any of the server computers in the Siebel Enterprise Server or on computers in an independent file server farm. For small-to-medium deployments, a common scenario might be to place the Siebel File System on the Siebel Gateway computer. Implementations with a large number of attachments might need a dedicated file system computer. Consider using a high-speed RAID disk storage system to increase file system throughput.
- It is strongly recommended that you disable short file-name generation on Windows servers hosting the Siebel File System, because this type of file-naming can cause severe performance impacts once the file system grows to a large size.
- During normal operation of Siebel CRM software, it is likely that orphaned files are stored in the Siebel File System and that orphaned records exist in the Siebel database. Periodically run the `sfscleanup` utility to remove orphaned files from the Siebel File System. This utility is located in the `bin` subdirectory within the Siebel Server root directory. For information about using this utility, see *Siebel System Administration Guide* .

About Siebel Server Load Balancing

Load balancing distributes workload across multiple Siebel Server computers. Each Siebel Server runs an instance of the service that you want to load balance. Load balancing also provides failover. If one Siebel Server fails, then requests are automatically routed to the remaining Siebel Servers.

The Siebel Gateway and Siebel Application Interface work together to provide Siebel Server load balancing. When a user requests a new application connection, Siebel Application Interface sends a request to Siebel Gateway, which returns a connect string for the least-loaded Application Object Manager from among the Siebel Servers supporting that component. The user session will use this Application Object Manager.

You can use load balancing when the Siebel Enterprise Server has two or more Siebel Servers that are not clustered. Load balancing is the preferred method for providing high availability for the following server components:

- Application Object Managers
- Siebel Product Configurator (uses own load balancing method)
- Siebel EAI (whenever possible)

On each Siebel Server, Siebel Connection Broker (SCBroker) provides intraserver load balancing. SCBroker distributes connection requests across multiple instances of Application Object Manager processes running on the same server computer. For more information, see *About the Siebel Connection Broker*.

Siebel CRM also supports load balancing across multiple installed instances of Siebel Application Interface. For more information, see *Recommendations for High Availability Deployments*.

About Siebel Internet Session Network API

Siebel Internet Session Network API (SISNAPI) is a Siebel proprietary message-body format running on top of TCP/IP. SISNAPI is used for communications between the Siebel Application Interface, Siebel Gateway, and Siebel Servers.

When the Siebel Application Interface receives a client request, it forwards it in SISNAPI format. The SISNAPI message-body format has the following parts:

- HTTP header
- Object Manager method name
- Method arguments as key-value pairs

For information about user request types, see *About User Request Types*.

HTTP Header

When the Siebel Application Interface requests a new connection, the initial packets of the first SISNAPI message contain an HTTP header. This header includes a Uniform Resource Locator (URL) that provides routing information to the Siebel Enterprise Server, Siebel Server, and server component.

Connection Multiplexing

SISNAPI TCP/IP connections are specific to an Application Object Manager on one Siebel Server. Before a new connection is opened, the Application Object Manager checks to see whether an existing connection is available. If so, then an existing connection is used. Once the connection is established, it remains open to be used by subsequent messages in the session or to be reused by other sessions. For more information about connection multiplexing, see *Siebel Performance Tuning Guide* .

Transport Layer Security (TLS)

SISNAPI connections use Transport Layer Security (TLS) for encryption and authentication. For more information, see *Siebel Security Guide* . See also *Siebel Installation Guide* .

User Request Types

The Siebel Application Interface generates three types of user requests. Each request type creates a new connection to a Siebel Server through the load balancer: initial request, retry request, and reconnect request. The Siebel native load balancing functionality is able to recognize and route these request types.

- **Initial request.** The Siebel Application Interface generates this request to start a new user session as follows:
  - The Siebel Application Interface receives the request to start a user session.
  - The Siebel Application Interface creates the message. The HTTP header specifies the Siebel Enterprise Server and the desired server component. The message does not specify a Siebel Server name.
  - The Siebel Application Interface sends the request message to Siebel Gateway.

    - The Siebel Gateway returns a connect string for a suitable component: the least-loaded Application Object Manager running on one of the Siebel Servers that support that component.
    - The Siebel Application Interface in turn routes the request to the Siebel Server where this Application Object Manager is running. If no connection exists to the Siebel Server, then a new one is created.
    - The Siebel Server receives the message and creates a new user session. The Siebel Server forwards address information back to the Siebel Application Interface.
    - The Siebel Application Interface creates a cookie containing the address information. The Siebel Application Interface receives the cookie information in subsequent session requests. The Siebel Application Interface includes this information in the HTTP header.
    - The load balancer receives subsequent messages and forwards them directly to the specified Siebel Server and server component through the open connection.
- **Retry request.** If a Siebel Server rejects an initial request, then the request is routed back to the Siebel Application Interface and the following occurs:

    - The Siebel Application Interface modifies the URL contained in the HTTP header by appending the letters *RR* to it.
    - The Siebel Application Interface resends the request to the Siebel Gateway for a suitable component on a different Siebel Server.
- **Reconnect request.** Siebel Application Interface generates a reconnect request when it receives a user request for an existing user session that does not have a SISNAPI connection. The Siebel Application Interface uses the session cookie information to include the Siebel Server address in the HTTP header.

  The reconnect request opens a new connection. Reconnect requests can occur for several reasons:

    - The SISNAPI connection was opened by Application Interface 1, but load balancing routes subsequent session messages to Application Interface 2, which does not have an existing connection.
    - The connection timeout is exceeded and the connection is closed.
    - The network environment closes the connection, for example due to a firewall time-out.

See also *About Siebel Internet Session Network API*.

About the Siebel Connection Broker

The Siebel Connection Broker (SCBroker) server component provides intraserver load balancing. SCBroker distributes server requests across multiple Application Object Manager processes running on a Siebel Server.

SCBroker listens on a configurable, static port for new requests. When a new request is received, it forwards the request to the Application Object Manager process with the least number of running tasks, or forwards the request to another Application Object Manager process in round-robin fashion. A user session is created on this Application Object Manager. Thereafter, the requests that apply to this session are directly sent to the Application Object Manager process hosting the session.

SCBroker is enabled by default and has several parameters:

- **PortNumber.** Sets the port number on which SCBroker listens. The default is 2321, but you can change the port number.
- **DfltTasks.** Sets the default number of processes for SCBroker. The recommended value is 2.

- **MaxTasks.** Sets the maximum number of processes for SCBroker. The recommended value is 2. This value cannot be less than DfltTasks.
- **AutoRestart.** Default is On. If SCBroker terminates abnormally, then this setting allows it to restart automatically. Setting this parameter to Off or False is not recommended.
- **ConnForwardAlgorithm.** The connection forwarding algorithm is the routing scheme for SCBroker to use when routing intraserver requests to Application Object Manager processes. Possible values are LL (least-loaded) and RR (round-robin). SCBroker uses the least-loaded algorithm by default.
  - **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 the ConnForwardAlgorithm parameter, see the information that follows this list of parameters.
- **ConnForwardTimeout.** The connection forward time-out determines how long SCBroker waits for an Application Object Manager process to accept a request. The default is 500 milliseconds. This time-out minimizes wait time when SCBroker forwards a connection request to an Application Object Manager process and the request cannot be accepted. If a time-out occurs, then SCBroker reports an error back to the Siebel Application Interface.
- **ConnRequestTimeout.** The connection request time-out determines how long SCBroker waits for all of the packets in an incoming new request. The default is 500 milliseconds. This time-out minimizes SCBroker wait time when TCP/IP requests are incomplete. If a time-out occurs, then the request is sent back to the Siebel Application Interface.

More Information about the ConnForwardAlgorithm Parameter

As noted, you can use the ConnForwardAlgorithm parameter to specify either least-loaded or round-robin routing. The round-robin routing scheme does not take into account the number of running tasks for each Application Object Manager process. It simply forwards each request to the next Application Object Manager process among the available processes. The least-loaded algorithm considers only the current number of tasks for each Application Object Manager process that is running.

The Siebel Gateway provides the connect string for the Siebel Application Interface to use for sending each request to an Application Object Manager process. If Siebel Gateway fails to provide a connect string, then Siebel Application Interface resends the request to the Siebel Gateway up to five times, with five seconds between each request.

In many circumstances, how an individual connection request is forwarded might be the same for either forwarding algorithm. The actual forwarding behavior depends on these factors:

- The setting of the ConnForwardAlgorithm parameter for SCBroker.
- The settings of the ConnForwardTimeout and ConnRequestTimeout parameters for SCBroker.
- The current number of Application Object Manager processes.
- The current number of tasks for the Application Object Manager component and for each Application Object Manager process.

- Patterns of requests for new tasks and patterns of threads being freed up through logouts.
- Application Object Manager parameter settings controlling the maximum number of tasks and the minimum and maximum number of Application Object Manager processes. The applicable parameters are:
    - Maximum Tasks (MaxTasks)
    - Minimum MT Servers (MinMTServers)
    - Maximum MT Servers (MaxMTServers)

  For information about calculating the settings for MaxTasks, MaxMTServers and MinMTServers, see *Siebel Performance Tuning Guide* .

Sometimes an individual connection request is not forwarded to an existing Application Object Manager process but instead causes a new Application Object Manager process to start. This can occur with either forwarding algorithm, when each existing Application Object Manager process has reached its theoretical maximum number of tasks, based on dividing the maximum number of tasks by the maximum number of Application Object Manager processes. A new process starts only if the maximum number of tasks and the maximum number of Application Object Manager processes have not yet been reached.

For more information about applicable parameters, see also *Siebel System Administration Guide* and *Siebel Performance Tuning Guide* .

About the Server Request Broker

The Server Request Broker (SRBroker) server component processes both synchronous and asynchronous server requests.

- Synchronous server requests are requests that must be run immediately, and for which the calling process waits for completion.
- Asynchronous server requests are requests for which the calling process does not wait for completion.

SRBroker can run server requests on any Siebel Server in the Siebel Enterprise. For example, if SRBroker is unable to run a server request on the local Siebel Server because the required component is not enabled, then SRBroker finds another Siebel Server that is hosting the required component and runs it there. SRBroker runs by default on all of the Siebel Servers.

SRBroker decides where to run a server request using the following criteria:

- If the required component is available locally, then SRBroker runs the task locally.
- If the required component is not available locally, then SRBroker identifies any Siebel Servers in the same Enterprise that have the component online. Server requests are submitted to each of these Siebel Servers in turn (a round-robin algorithm).
- If the required component is not available anywhere in the Enterprise, then the server request fails.

The SRBroker component helps provide resilient processing. As long as the required component is running on a Siebel Server somewhere in the Enterprise, then the server request can be processed. For more information about resilient processing, see *About Resilient Processing*.

About the Server Request Processor

The Server Request Processor (SRProc) server component processes asynchronous, server-initiated requests. These are requests that are submitted for later execution and that do not require the calling process to wait for the request to complete.

SRProc runs by default on all of the Siebel Servers. When asynchronous requests are submitted, they are stored in the Siebel database in the S_SRM_REQUEST table. SRProc periodically checks this table for any requests that are eligible to be run. For a request to be eligible, it must meet all of the following criteria:

- The request must be in the correct state (Queued)
- Its start time must have passed
- The target Siebel Server must not be specified or must be the Siebel Server where the requested component is running

If a request is eligible, then SRProc invokes Server Request Broker (SRBroker) to run the request, as described in *About the Server Request Broker*. Therefore, as long as a target Siebel Server is not specified, asynchronous requests are read by any SRProc task on any Siebel Server.

The SRProc component helps provide resilient processing for server-initiated tasks. As long as an SRProc task is running somewhere in the Siebel Enterprise, the request is processed. For more information about resilient processing, see *About Resilient Processing*.

About Siebel Enterprise Application Integration

Siebel Enterprise Application Integration (EAI) provides components for integrating Siebel CRM with external applications and technologies. It is designed to work with third-party solutions such as those from IBM, CrossWorlds, TIBCO, Vitria, SeeBeyond, webMethods, and others.

Siebel EAI provides bidirectional real-time and batch solutions for integrating Siebel applications with other applications.

Siebel EAI is designed as a set of interfaces that interact with each other and with other components within the Siebel application. These interfaces are compatible with IBM MQSeries, Microsoft MSMQ, Java and Java EE, XML, HTTP, and many other standards.

Siebel EAI interfaces do the following:

- Allow a flexible service-based architecture built on top of configurable messages using XML and other formats.
- Expose internal Siebel objects to external applications
- Take advantage of prebuilt adapters and enterprise connectors, and are compatible with third-party adapters and connectors.
- Allow for data transformation.
- Integrate external data through virtual business components (VBCs).
- Provide a graphical business process designer, programmatic interfaces, and a high-volume batch interface.

For more information about Siebel EAI, see *Overview: Siebel Enterprise Application Integration* and other applicable documentation on *Siebel Bookshelf*.

About Siebel Enterprise Integration Manager

Siebel Enterprise Integration Manager (EIM) manages the bidirectional exchange of data between the Siebel database and other corporate databases. This exchange is accomplished through intermediary tables called EIM tables. The EIM tables act as a staging area between the Siebel database and other databases.

You must use Siebel EIM to perform bulk imports, exports, updates, and deletes. Using native SQL to load data directly into Siebel base tables (the tables targeted to receive the data) is not supported.

For more information about Siebel EIM, see *Siebel Enterprise Integration Manager Administration Guide* .

About Siebel Tools and Siebel Web Tools

Siebel CRM provides two main development tools:

- Siebel Tools is a Windows-based, integrated environment for configuring Siebel applications.
- Siebel Web Tools is an Application Object Manager component that provides similar functionality through the Siebel Web Client. Most of the information provided about Siebel Tools also applies to Siebel Web Tools.

You use Siebel Tools to modify standard Siebel objects and create new objects to meet your organization's business requirements. For example, you use Siebel Tools to extend the data model, modify business logic, and define the user interface. Siebel Tools is also a way to integrate programs written using Siebel scripting languages.

A standard Siebel application provides a core set of object definitions that you can use as a basis for your own tailored application. Siebel Tools object definitions are grouped into four layers, each with a different purpose:

- **Physical user interface (UI) layer.** Templates and tags that render the user interface in the client.
- **Logical user interface objects layer.** Presentation of data (user interface).
- **Business objects layer.** Objects that extract defined information from the database or provide a defined service.
- **Data objects layer.** Database interface objects and table definitions.

Object types in a given layer depend on definitions in the next lower layer, and are insulated from other layers in the structure. You can make certain kinds of changes to a Siebel application without changing the underlying database structure. Similarly, you can extend the Siebel database schema without affecting the Siebel application. In many cases, configuration changes are made in concert across multiple layers in order to achieve the desired business functionality.

As of Siebel CRM 17.0, Siebel CRM applications use the Siebel runtime repository in the Siebel database.

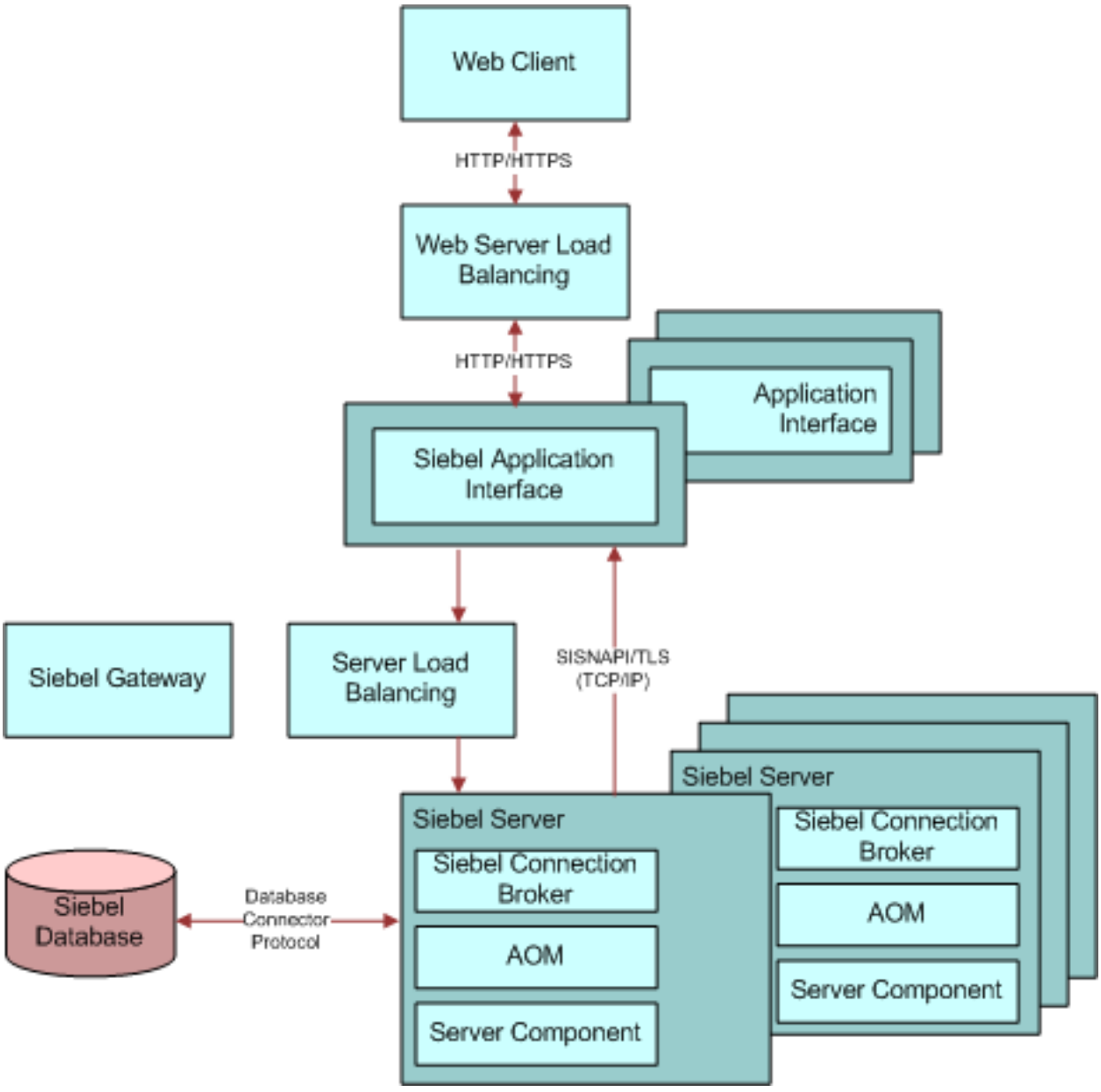
For information about using Siebel Tools and Siebel Web Tools, see *Using Siebel Tools* , *Configuring Siebel Business Applications* , and other guides. For Siebel Tools installation instructions, see *Siebel Installation Guide* .

Many aspects of the physical user interface can also be configured outside of Siebel Tools. For more information, see *Configuring Siebel Open UI* .

Example of User Request Flow in a Siebel CRM Deployment

The following figure illustrates how a user request is processed within the Siebel CRM architecture. In the diagram, there are two types of load balancing:

- **Siebel Application Interface load balancing.** Web client requests can be forwarded to different instances of Siebel Application Interface. This type of load balancing is also called Web server load balancing.
- **Siebel Server load balancing.** Siebel Application Interface and Siebel Gateway work together to distribute user requests to multiple Siebel Servers.



A typical Siebel client request flows from the user's Siebel Web Client through the system components for Siebel CRM applications and back again, according to the following general flow:

1. A user performs an action that initiates a request. For example, the user clicks a link in the Site Map to navigate to a particular view. The request is generated by the Web browser and Siebel Web Client framework.
2. The request goes through the network, using an existing or new HTTP connection. The request might go through a network router, proxy server, cache engine, or other mechanism.
3. The Web server load balancer, if present, evaluates the request, determines the best Siebel Application Interface to receive the request, and then forwards the request to the Siebel Application Interface.
4. The Siebel Application Interface receives the HTTP request and determines that it is a Siebel application request.
5. The Siebel Application Interface parses the HTTP message and generates a SISNAPI message, based on the content of the HTTP message. The Siebel Application Interface also parses the incoming cookie to obtain the user session ID. The Siebel Application Interface forwards the request to a Siebel Server on a least-loaded basis.

   > **Note:** Siebel CRM applications require session cookies and do not support cookieless mode.

6. On the Siebel Server, an Application Object Manager (AOM) receives and processes the SISNAPI message. If a database query is needed to retrieve the information, then the Application Object Manager formulates the SQL statement and sends the request to the Siebel database over a database connection.

   The database request uses a protocol format that is specific to the database connector.
7. The database executes the SQL statement and returns data back to the Application Object Manager. The Application Object Manager forwards the message to the Siebel Application Interface that originated it.
8. The Siebel Application Interface receives the SISNAPI message and translates it back to HTTP. The message is now in the form of Web page content.
9. The Siebel Application Interface forwards the Web page content through the original HTTP connection to the end user's Web browser.
10. The Web browser and the Siebel Web Client framework process the return message and display it to the end user.

About Siebel Open UI

Siebel CRM applications use Siebel Open UI. Siebel Open UI provides a rich user interface experience and is based on browser standards supported by recent versions of several browsers.

For more information about configuring and deploying Siebel CRM applications with Siebel Open UI, see the following related documentation:

- Applications previously deployed using high interactivity now use Siebel Open UI. Many applications previously deployed using standard interactivity have been updated to use Siebel Open UI, while others might no longer work. Migrate your users to supported employee and customer applications, as appropriate. For more information, see *Siebel Installation Guide* and other relevant documentation.
- For information about deploying Siebel CRM applications for your Web clients, see *Deploying Siebel Open UI* .
- For information about configuring Siebel Open UI features for Siebel CRM applications, see *Configuring Siebel Open UI* .
- For information about using Siebel CRM applications, see *Siebel Fundamentals Guide* .
- For information about the functionality that is available for Siebel Open UI, see *Siebel Open UI Deployment Guide* on My Oracle Support (Article ID 1499842.1).

- For the minimum browser standards for Siebel Open UI, see the Certifications tab on My Oracle Support.

3 Siebel Infrastructure Planning

Siebel Infrastructure Planning

This chapter explains how to plan the infrastructure of your Siebel CRM deployment. It includes the following topics:

- *Process of Infrastructure Planning*
- *Determining How Siebel CRM Applications Are Used*
- *Defining Data Flows and Integration Requirements*
- *Determining Database Requirements*
- *Mapping Business Requirements to Siebel Server Components*
- *Defining High Availability Policies*
- *Mapping Siebel CRM Deployment Elements to Platforms*
- *Determining Network Requirements*
- *Defining a Test and Transition Plan for the Siebel CRM Deployment*

Process of Infrastructure Planning

The tasks in this process help you to determine the Siebel CRM infrastructure requirements for a production environment. Along with a production environment, you must also plan for a development environment and a test environment.

Use the following tasks to plan your Siebel CRM deployment infrastructure:

1. *Determining How Siebel CRM Applications Are Used*
2. *Defining Data Flows and Integration Requirements*
3. *Determining Database Requirements*
4. *Mapping Business Requirements to Siebel Server Components*
5. *Defining High Availability Policies*
6. *Mapping Siebel CRM Deployment Elements to Platforms*
7. *Determining Network Requirements*
8. *Defining a Test and Transition Plan for the Siebel CRM Deployment*

Determining How Siebel CRM Applications Are Used

This task is a step in *Process of Infrastructure Planning*.

This infrastructure planning task identifies what tasks users perform when using Siebel CRM applications. Examples are completing a customer order, adding a contact, and creating a quote. Later in the planning process, you map these tasks to specific Siebel applications and functions.

To determine how Siebel CRM applications are used

1. Identify user types.

   For each business location, identify user types. Organize this list by the functional areas that participate in key business processes. Include application developers and integrators, system administrators, and application administrators in your list of user types.

   For example, assume that you have a call center in Denver. One of your key business processes is order creation. Two of the functional areas that participate in this business process are call center agents and product line administrators. These are two user types.
2. Identify tasks by user type.

   For each user type, identify all of the tasks that the users perform using Siebel applications. Start with each key business process and map its steps to tasks. Doing this step helps you to verify that your business processes are being correctly automated.
3. Identify background tasks.

   If your business operation includes background tasks, then list these as well. Background tasks are those that the Siebel CRM components perform, rather than users. These include batch processing of business data and automated workflow processes.
4. Estimate transaction volumes.

   For each user task, estimate average and maximum daily transaction volumes. For example, in your Denver call center there are 25 call center agents. Transaction records indicate that each agent completes an average of 12 customer orders per day and a maximum of 20 per day. The following is an example of how you would list transaction volumes for the Denver call center.

User Type	Number	Task	Average Volume Per Day	Maximum Volume per Day
Call Center Agents	25	1. Inbound customer order	300	500

Defining Data Flows and Integration Requirements

This task is a step in *Process of Infrastructure Planning*.

This infrastructure planning task identifies how data flows to and from the Siebel CRM deployment. An example of a key data flow would be customer contact updates that originate at several call centers and flow to the master customer contact database at a headquarters location.

This task identifies where the master copy of data records resides. It also identifies the data interchange requirements for applications.

To identify data flows and transaction volumes

1. Identify business data.

List the types of business data that flow through the system components of Siebel CRM. Examples of business data are orders, customer contacts, product line information, and quotes.

2. Identify business data sources.

   For each type of business data, list the user types or business activities that can originate or update the business data. Group user types or business activities by business location.

3. Analyze the data requirements of legacy applications.

   Identify all of the existing applications that send or receive data from the Siebel CRM deployment. Determine data volumes and group them by location.

4. Identify data formats and transformations.

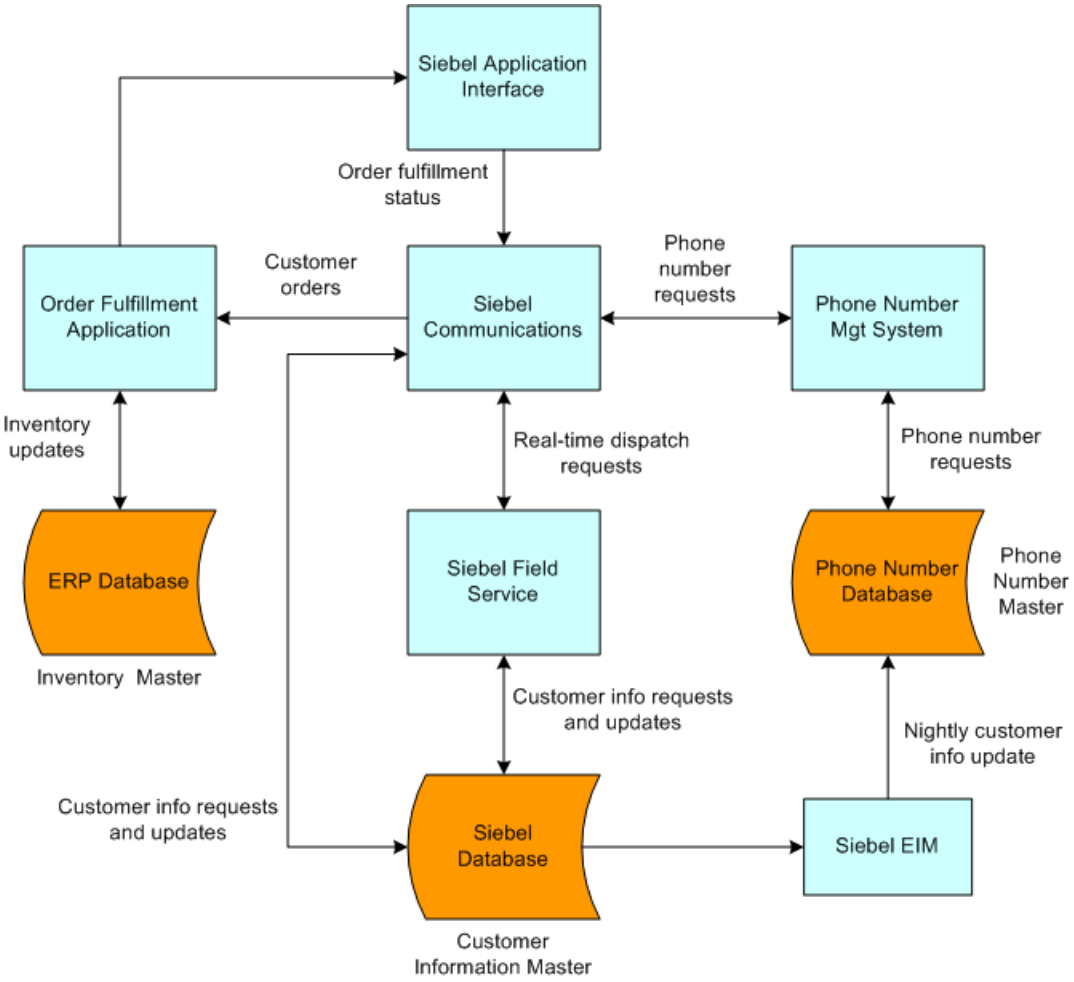
   For each legacy application that sends or receives data from the Siebel application, identify the required data formats. Specify in detail all of the data transformation requirements.

5. Map the data flows.

   Create a model that shows all of the major business data flows. Include all of the data sources, repositories, and key business applications.

The following figure shows an example of a model of a data flow. The example shows a call center running the Siebel Communications application. The company maintains an ERP database and a phone number database separately from the Siebel database, which contains customer information.

Siebel Communications sends XML messages containing customer orders to the order fulfillment application, and receives order fulfillment status through an inbound HTTP adapter. Siebel Communications also queries the phone number database for available phone numbers in real time. The phone number database then receives assigned phone numbers from the Siebel database using Siebel EIM.



Determining Database Requirements

This task is a step in *Process of Infrastructure Planning*.

This infrastructure planning task identifies database requirements for the Siebel CRM deployment.

You would have already identified the types of data that are stored in the Siebel database. This task maps that data to key database characteristics. Doing this task helps you to estimate database size requirements and expected growth. Begin by defining general requirements:

- What types of records are stored? What specific fields does each record contain?
- What is the volume of each record? How many records of each type are processed each hour? Each day? Each year? Group this information by business location.
- Determine how record volumes map to specific Siebel tables. For help with mapping records to Siebel tables, contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services.
- How much space do database indexes occupy? Typically, indexes require as much space as the data. For example, 50 GB of data requires about 50 GB of indexes.

- What is the expected annual growth rate of the database by record type and location?

Include the following information in your analysis of records:

- Number of addresses that are assigned to each customer account
- Number of employees that are assigned to each account
- Number of contacts that are assigned to each account
- Number of attachments that are assigned to a record
- Number of activities that are associated with each account
- Whether opportunities, quotes, or orders are stored
- Whether product data is stored
- Whether Siebel Remote is used
- Whether Siebel Mobile applications are used

Include the temporary table space, log files, and space required for loading data.

Mapping Business Requirements to Siebel Server Components

This task is a step in *Process of Infrastructure Planning*.

This infrastructure planning task identifies the Siebel Server components needed to meet your business requirements.

Begin by listing the Siebel applications that users run. For each application, identify the associated Application Object Manager. List all language-specific Application Object Managers that you need.

Many Application Object Managers require additional server components such as Workflow Manager. Users typically do not interact with these second-tier components directly. The role of these components is to support the function of Application Object Managers and of the Siebel Server.

All of the second-tier component requirements must be correctly identified in your planning and review phases. Work closely with your implementation team to identify these components.

After you have identified all of the required server components, group them by business location. Then, for each location, determine the anticipated workload volumes for all of the components. Consider both average and peak workloads. This information is key for deciding how to distribute Application Object Managers and other components across Siebel Servers.

Defining High Availability Policies

This task is a step in *Process of Infrastructure Planning*.

This infrastructure planning task defines business policies regarding availability of servers.

See also *High Availability Deployment Planning* .

Siebel Servers

For each business location, assess the impact of losing each server component. Consider the possibility of the component failing, rather than the hosting platform itself. Individual server components that are important to normal application function must be identified in your planning and review phases. Work closely with your implementation team to identify all of the components that could represent single points of failure.

After you complete this analysis, define high availability policies for all of the applications and services. Decide how long your business can tolerate not having access to key applications. Also, decide how long your business can tolerate degraded performance.

For example, a company decides that Siebel Call Center must run 24 hours a day, seven days a week, and that the maximum acceptable downtime is 30 minutes. The company also decides that the maximum time it can accept degraded performance is one hour.

Finally, at each business location, list all of the server components to which each policy applies. This analysis forms the basis for implementing a high availability strategy as part of hardware planning.

Database Platform and Data Integrity

The server platform that hosts the Siebel database is crucial to Siebel CRM deployment operations. For this reason, it is important to define high availability and data integrity policies specifically for the database server. The following policies are recommended:

- Cluster database servers to protect against platform hardware failures.
- Use redundant disk arrays (RAIDs) for disk storage. RAID 1+0 is recommended because it provides maximum performance, and there is no data loss if a disk fails. Do not implement RAID 0 arrays. RAID 0 offers good performance but does not protect data adequately in the event of a disk failure.
- Enable transaction logging.
- Observe the following recommendations for storing database files:
    - Store data and indexes on separate disk subsystems.
    - Store active log files and archived log files on separate disk subsystems.
    - Store the database and database control files on separate disk subsystems.
- To allow for good OLTP performance, set up four rollback segments (if you choose to use them) for each 20 to 40 users. For Initial extents or Next extents, set up rollback extents sized to 100 KB. If you use Siebel EIM, then also create several additional, large rollback segments to support Siebel EIM loads.

Siebel Gateway

The Siebel Gateway maintains the Siebel Gateway registry to store the configuration information for all of the Siebel Servers in all of the Siebel Enterprise Servers managed by the Siebel Gateway. Loss of the Siebel Gateway due to a disk failure could bring your Siebel CRM deployment to a halt while you restore the Siebel Gateway, unless you take measures to provide high availability for the Siebel Gateway.

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. This feature works at the software level and is the preferred and recommended approach for clustering the Siebel Gateway. To use this feature, you install Siebel Gateway on multiple nodes and configure clustering using Siebel Management Console. Clustering is supported for both the Siebel Gateway service (application container) and the Siebel Gateway registry (Apache ZooKeeper). If an individual node goes down within the Siebel Gateway cluster, then the Siebel Application Interface or Siebel Server client connection switches to

another available node. After any Siebel Gateway cluster node goes down, when it is restarted, the node will again participate in the cluster. For more information, see *Siebel Installation Guide* .

Optionally, you can install a RAID or some other type of redundant disk configuration on your Siebel Gateway.

Siebel Mobile Web Client Users

A Siebel Server temporarily stores transaction files that move to and from users of the Siebel Remote client, generally called Siebel Mobile Web Clients. The loss of these files results in the need to re-extract the database for all of the affected mobile users. (Siebel Remote supports synchronization of data between Siebel Mobile Web Clients and the Siebel database, over an Internet connection.)

It is strongly recommended that you install a RAID or some other type of redundant disk configuration on Siebel Servers that run Siebel Remote.

Mapping Siebel CRM Deployment Elements to Platforms

This task is a step in *Process of Infrastructure Planning*.

This infrastructure planning task maps the elements of the Siebel CRM deployment to server platforms.

This topic includes the following information:

- *Determining Server Platform Requirements*
- *Topology Planning Guidelines*
- *Recommendations for Collocating Server Components*

Criteria

Mapping Siebel CRM deployment elements to platforms must meet the following criteria:

- Guarantees adequate performance and scalability under both average and peak workloads
- Meets high availability and resiliency goals
- Accommodates infrastructure security requirements

Requirements

Review the following information, developed in previous tasks:

- **Database requirements.** See *Determining Database Requirements*.
- **Required Siebel Server components.** See *Mapping Business Requirements to Siebel Server Components*.
- **High availability policies.** See *Defining High Availability Policies*.

Determining Server Platform Requirements

This topic is part of *Mapping Siebel CRM Deployment Elements to Platforms*.

Perform the following steps to determine your server platform requirements, for topology planning:

To determine server platform requirements

1. Determine the amount of hardware required for Siebel Server components. Consider both average and peak workloads. Also consider background processing workloads.

   On two- or four-CPU platforms, customers typically deploy one Application Object Manager on each Siebel Server. Deploying Application Object Managers in this manner is a common practice, but not a requirement. Depending on the number of concurrent users, the amount and complexity of customization and the components used, the distribution of components can vary.

   For information about calculating the settings for MaxTasks, MaxMTServers, and MinMTServers, see *Siebel Performance Tuning Guide* .

   For additional help with sizing Siebel Servers, contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services.
2. Identify which Siebel Server components you can collocate. Distribute these across platforms in a way that evenly distributes workload.

   Recommendations for collocating server components are provided later in this topic.
3. Determine how many additional hardware platforms are needed to comply with high availability policies.

   For clustered servers, define a failover strategy for components (active-active, active-passive).

   > **Note:** In active-active clustering, a process is only active on one node of the cluster and is not active on the other node; that is, two physical servers are running a different clustered process. For information about high availability options, see *Defining High Availability Policies*.
4. Identify additional hardware required to comply with security policies. For example, do you have to install additional firewalls or a proxy server? Do you have to install LDAP servers?
5. Use average and peak workload information to determine how many instances of Siebel Application Interface are needed.
6. Create a diagram of the Siebel CRM deployment that shows all of the platforms and the distribution of Siebel Servers. Use the diagram to do the following:

   a. Verify that all of the needed server components are enabled and correctly set up on each platform.
   b. Run component and platform failure scenarios. Verify that there are no single points of failure that can cause unacceptable impacts.

   For example, assume that you have one Siebel Application Interface. All of your inbound customer orders must go through it and then to one HTTP inbound adapter. If the Siebel Application Interface or the inbound adapter fails, then customers cannot place orders.
7. Use server naming conventions to identify groups of servers that provide similar functions.

   For example, in an enterprise, Application Object Managers run on one group of computers, workflows on a second group of computers, and remote user synchronization on a third group. Give the Application Object Manager servers names starting with APP, the workflow servers names starting with WF, and the Siebel Remote servers names starting with REM.

   The servers in each group are displayed together in Server Manager, which simplifies server administration.

   > **Note:** Additional considerations for server naming are provided in *Siebel Installation Guide* .

Topology Planning Guidelines

This topic is part of *Mapping Siebel CRM Deployment Elements to Platforms*.

Consider the following guidelines for topology planning:

- A single Siebel Gateway or Siebel Gateway cluster can manage a single Siebel Enterprise Server.
- A Siebel Enterprise Server can belong to one and only one Siebel Gateway.
- A single Siebel Enterprise Server can manage multiple Siebel Servers.
- A Siebel Server can belong to one and only one Siebel Enterprise Server.
- A Siebel Server can manage multiple instances of a single server component or of multiple server components. Components can include multiple Application Object Manager types, each with its own Siebel runtime repository.
- The Siebel Servers in a Siebel Enterprise Server can connect to only one Siebel database.

The following table describes possible deployment schemes for Siebel CRM applications.

Deployment Scheme	Recommended?
A single Siebel Gateway with multiple Siebel Enterprise Servers configured on a single computer or UNIX hardware partition. Each Siebel Enterprise Server has its table owner and Siebel database.	No.
Running multiple Siebel Gateways on a single UNIX hardware partition or on a single unpartitioned computer.	No.
Installing and configuring Siebel Gateway on multiple nodes to provide clustering services.	Yes. For more information, see *Defining High Availability Policies*. See also *Siebel Installation Guide* .
Multiple Siebel Enterprise Servers sharing a DBMS table owner (schema).	Might be suitable for some deployments. There are many possible scenarios for deploying with multiple Enterprise Servers and many issues to consider. For more information, see 477829.1 (Article ID) on My Oracle Support. This document was previously published as Siebel Technical Note 544.
Multiple Siebel Enterprise Servers, each with their own DBMS table owner and sharing the same instance of the DBMS executable.	No.
A single Siebel Enterprise Server hosting multiple Siebel Servers within a single hardware partition.	No.
A Siebel Enterprise Server hosting multiple Siebel Servers, with each Siebel Server	Yes. This scheme is the most common way to deploy Siebel CRM applications.

Deployment Scheme	Recommended?
on its own computer, operating system instance, or UNIX hardware partition (multiple partitions on each UNIX server computer).	
Installing the Siebel Gateway, each Siebel Server, and the Siebel database all on different operating systems.	Yes. This scheme is supported. However, it is strongly recommended to keep the deployment as simple as possible. In some cases a heterogeneous environment is required. For example, you want to install Siebel Servers running on one operating system, but a third-party product that you need only runs on another operating system. For information about supported operating systems for Siebel CRM deployments, see the Certifications tab on My Oracle Support.

Recommendations for Collocating Server Components

This topic is part of *Mapping Siebel CRM Deployment Elements to Platforms*.

When deciding which server components to collocate, consider the following recommendations:

- You use the Siebel Enterprise Server installer to install all of the server modules for your Siebel CRM applications deployment. Many installation charactistics have changed significantly as of Siebel CRM 21.2 Update, as described in *Siebel Installation Guide* . Siebel Server, Siebel Gateway, Siebel Application Interface, Siebel Enterprise Cache, and other server modules are always installed for any installation of Siebel Enterprise Components. Installed modules are available to be configured and deployed using Siebel Management Console, as you choose. You must keep track of which installed modules you are configuring and deploying, and where they are deployed.

  You might choose to deploy some of these modules together as part of the same installation (collocate them), install and deploy modules in separate installation locations on the same computer or operating system instance, or install and deploy them on different computers. In production environments, and particularly in large deployments, it is generally recommended to deploy modules separately, whether on the same computer or on different computers. The different deployment options present different benefits and risks, particularly related to availability and reliability in the event of computers requiring restart, or of shared application containers requiring restart. Some decision factors for collocating Siebel Server components or component groups are also noted in this topic.
- Where Siebel Server and Siebel Gateway are installed and deployed together (collocated), they share the same application container. If the application container is shut down or restarts, then this affects all modules sharing the application container, severely limiting availability and reliability for these modules. Siebel Application Interface uses its own application container even where it is collocated with Siebel Server or Siebel Gateway.

  Assume, for example, that Siebel Server and Siebel Gateway are collocated. If, on the Siebel Server, there are issues with outbound REST or SOAP on UNIX, JBS (Java Business Services), or inbound email using the email servlet, or if there are heavy loads for these features, then restarting the application container can impact the Siebel Gateway or Siebel Web Clients. Installing, configuring, and removing Siebel Servers also use the application container. Note, however, that restarting the application container does not present issues where Siebel Server and Siebel Gateway are not collocated, or where these modules are collated but where operations

of these types do not apply. Customers must consider such factors, among others, when planning their Siebel CRM deployment topology and before installing and deploying Siebel CRM server modules.

- As you plan your Siebel CRM topology, you must also plan for the input that you must provide when you run the Siebel CRM installer or run Siebel Management Console, including data such as port numbers used by Siebel modules. You must anticipate various installation and configuration requirements. You must have a security and authentication framework in place to be able to install and configure the Siebel software. After your installations are done, you must install the Siebel database and configure various installed Siebel modules. Using Siebel Management Console, you can configure the Siebel Gateway, Siebel Enterprise, Siebel Server, Siebel Application Interface, and other modules. The Siebel Gateway, Siebel Server, and Siebel Application Interface need to be able to communicate with one another in order to configure and operate your installed software.

  For more information, see *Siebel Installation Guide* and *Siebel Security Guide* .
- Install the Siebel Document Server on a dedicated server. Siebel Document Server uses Microsoft Word to create documents. Because Word is a single-threaded process, Siebel Document Server could block other processes running on the same server.
- Consider what time of day a component is used. For example, a typical scenario is to run Siebel EIM batch jobs during off-peak hours. Doing this means Siebel EIM can be collocated with a server that is busy during peak hours. You can collocate components running off-peak with on-peak components for server consolidation purposes.
- Collocation architecture decisions are driven by the load placed on each component and the overall size of the deployment. Consider this scenario: an implementation plan involves having 1000 connected users with light Assignment Manager usage, light Siebel Workflow usage, heavy off-peak Siebel EIM batch jobs, and a moderate Communications Server load. This implementation also requires some degree of high availability and resiliency. One possible architecture solution would be as follows:
  - Clustered Siebel Gateway. Collocated Siebel File System, Assignment Manager, Workflow Monitor Agent, and Communications Server. You can also configure Siebel Gateway clustering using Siebel Management Console, as noted in *Defining High Availability Policies*.
  - Multiple, load-balanced Siebel Application Object Managers. The number of servers required depends on the level of configuration and scripting complexity.
- You can choose to engage Oracle's Application Expert Services to perform an architecture and sizing review early in the implementation process. Once key metrics are known (user count, data volume, transaction volume, interfaces, and so on), you can determine the architecture and size of the Siebel CRM deployment. Contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services.
- Some components are complex to size (for example, Siebel Product Configurator) and require expertise to determine an appropriate architecture and distribution of components. It is always necessary to have a thorough understanding of your customizable products and how they are used.
- It is common practice in a large implementation (with thousands of users) to dedicate a server (usually a set of servers) to one function. Doing this makes it easier to monitor the performance of each segment of the implementation and makes it easier to scale each subset of the architecture. In a large implementation, Application Object Managers, Workflow, EAI, Assignment Manager and Siebel Product Configurator would all run on dedicated servers. Instances of Siebel Gateway and Siebel Application Interface would also run on dedicated hardware.
- When there are more than a relatively small number of remote users (100 or so), run server components from the Siebel Remote (alias Remote) and Disconnected Mobile Synchronization (alias MobileSync) component groups on a dedicated server. Planning considerations must include the total number of remote users, the expected data volume transferred between the enterprise database and the remote client, and the quantity of visibility events.

Visibility events include adding or removing a position to an account team, adding or removing a user to a position, and so on. When a visibility event occurs, it can cause a great deal of Siebel Remote activity.

- In general, it is important to identify those processes with the potential for spikes of resource consumption and spread the load accordingly.

Related Books

Siebel Installation Guide

Siebel Security Guide

Determining Network Requirements

This task is a step in *Process of Infrastructure Planning*.

The purpose of this infrastructure planning task is to identify the network requirements needed to support the Siebel CRM deployment.

To determine network requirements

1. Use the information about average and peak workloads to verify that there is sufficient network bandwidth to handle network traffic to and from, as well as within, the Siebel CRM deployment.
2. Determine whether to use data encryption. If so, then define data encryption policies. Then add data encryption protocols to the deployment plans that you outlined using *Mapping Siebel CRM Deployment Elements to Platforms*.
3. Define firewall requirements. If you are creating a network DMZ, then also define the requirements for proxy servers and other items that you install in the DMZ.

   Include network address translation (NAT) and HTTPS requirements.
4. Analyze interactions and dependencies between networking components.

   For example, assume that you use HTTP with TLS (Transport Layer Security) between browsers and instances of Siebel Application Interface.
5. Select port numbers for all of the network components that listen on TCP ports.

   This includes Siebel Application Interface, Siebel Servers, and server clusters. For a full description of Siebel Server components that require a port number assignment, see *Siebel System Administration Guide* . See also *Siebel Installation Guide* .

   The default TCP port number for Siebel Application Interface-to-Siebel Server traffic is 3320. This is the port number of the Siebel Connection Broker (SCBroker); however, you can configure the port number. You cannot assign the Siebel Gateway and Siebel Servers port numbers higher than 32767.

   Also, verify that firewalls are configured to communicate with the correct TCP ports.
6. Consider any other factors that might affect networking connectivity.

Defining a Test and Transition Plan for the Siebel CRM Deployment

This task is a step in *Process of Infrastructure Planning*.

It is important that you define a test plan to verify that the proposed deployment infrastructure functions correctly and is sized correctly. Equally important is defining a plan that transitions the Siebel CRM deployment to production.

Observe the following recommendations for testing the Siebel CRM deployment and transitioning it to production:

- **Separate production environment.** Keep the development and test environments physically separate from the production environment. Do not conduct development and test activities on the production Siebel database or, if possible, on the production database server.
- **Server stress testing.** Test Siebel Enterprise Server performance under average and peak workloads.

  Oracle's Application Expert Services finds that performance problems at customer sites are frequently caused by the following:

  - Servers were tested at much less than average or peak workloads. This prevents configuration and tuning problems from being uncovered.
  - Siebel Server components and other Siebel product modules are either incorrectly distributed across servers or are not configured correctly.
- **Failover and resiliency testing.** Define a test plan that evaluates the effect of server component failures. Work closely with your implementation team to identify all of the components that could represent single points of failure, as noted in *Defining High Availability Policies*.

  Define a server cluster test plan that evaluates failover behaviors. Run the test plan under average and peak workloads. It is particularly important to verify that failover performance under peak workloads is acceptable.
- **Database server testing.** Define a test plan that evaluates the following:

  - **OLTP performance.** Consider OLTP performance under average and peak workloads.
  - **Database server platform failover.** Typically the database server is clustered.
  - **Recovery from database corruption.** The database vendor usually provides recovery mechanisms.
  - **Batch processing support.** Verify that the database server correctly handles batch jobs from servers as well as synchronization requests from Siebel Remote components.
  - **Siebel Web Client users.** Verify that batch jobs do not degrade transaction processing performance and are completed in a reasonable time.

4 High Availability Deployment Planning

High Availability Deployment Planning

This chapter provides information about planning high availability for your Siebel CRM deployment. It includes the following topics:

- *How Service Failures Affect the Siebel CRM Deployment*
- *About High Availability Deployment Options*
- *Recommended High Availability Techniques for Specific Services*
- *Recommendations for High Availability Deployments*
- *About Resilient Processing*

See also *Defining High Availability Policies*.

How Service Failures Affect the Siebel CRM Deployment

This topic describes how major architectural components in a Siebel CRM deployment are affected when a service failure occurs. Such failures can be prevented or mitigated by using high availability deployment options. Services include both hardware platforms and software applications. Subtopics include:

- *Components Involved in Service Failures*
- *Impact of Service Failures*
- *Specific Failures and Associated Impact*

Components Involved in Service Failures

This topic identifies major architectural components in a Siebel CRM deployment that might be affected when a service failure occurs.

This topic is part of *How Service Failures Affect the Siebel CRM Deployment*.

Siebel Web Clients

Client hardware failure and browser failures are the most common causes of Siebel Web Client failure. Operating system failures can also cause this, but are rare.

When the Siebel Web Client fails, user sessions are lost even though the sessions might continue running on the Siebel Server for a time. The user session is lost because when the Siebel Web Client fails, the Siebel session cookie usually is also lost. Without the cookie, the user cannot be routed back to the existing user session on the Siebel Server. Therefore, the user usually has to log in again and start a new user session.

Siebel Application Interface

An Siebel Application Interface instance might fail because of hardware or software issues. When the Siebel Application Interface fails, Siebel Web Clients cannot access Siebel applications, because requests must go through the Siebel Application Interface first. Existing connections from the Siebel Application Interface to Siebel Servers are also lost.

If Siebel Application Interface is set up for high availability, for example if there are multiple instances of Siebel Application Interface, then subsequent requests can be routed to another working Siebel Application Interface. Usually when this occurs, the function of affected Siebel Web Client user sessions is not noticeably affected.

> **Note:** As of Siebel CRM version 17.5, application containers for Siebel Application Interface instances on multiple nodes can be load balanced using Apache HTTP Server (httpd) and Apache Tomcat Connector (mod_jk). If you configure load balancing in this way, then, whenever an application container fails for a Siebel Application Interface node, the active sessions handled by that node transparently failover to another available healthy application container node. For more information about Apache HTTP Server and Apache Tomcat Connector, refer to vendor documentation.

Siebel Servers

Siebel Servers might fail because of hardware or software issues. If the hardware platform fails, or the Siebel Server software fails, then all of the Siebel Server components are lost.

In other cases, individual Siebel Server components might fail. In turn, component failure can cause related user sessions or user requests to fail. The major groups of Siebel Server components are as follows:

- **Application Object Managers.** When Application Object Manager processes terminate unexpectedly, user sessions hosted by the Application Object Manager are lost. Users must log in to the Siebel application again. If users return to the same Siebel Server, then SCBroker tries to route the user request to a running Application Object Manager process.

  If there is only one Application Object Manager process and it has failed, then the request is directed to a different Siebel Server, unless there is only one Siebel Server.
- **Batch-mode server components going through SRBroker.** Most batch-mode server components receive server requests through SRBroker. An example is Workflow Manager. When a batch-mode component fails, the current server request fails.
- **Synchronous server requests.** An error is returned to the requesting component.
- **Asynchronous server requests.** An error is logged but not returned to the requesting component. Subsequent requests for the failed batch-mode component are attempted against a different instance of the component on the same Siebel Server or on a different Siebel Server.

  If no instance of the batch-mode component is available, then the request is logged to the S_SRM_REQUEST table to be processed later.
- **Direct Object Manager requests.** Examples of direct Object Manager requests are those to Siebel Product Configuration Object Manager. Some components, such as Siebel Product Configurator, have a native failover mechanism.
- **Other server components with location restrictions.** There are specialized server components that do not communicate through SRBroker. Siebel Remote Server is an example. Typically, requests to these components can only be processed by a specific Siebel Server. Therefore, if the server fails, then requests to that server fail, until the server is restarted.

Siebel Database

Access to the Siebel database can fail due to a number of factors:

- Database server hardware failure
- Database server running out of resources
- Disk failure
- Network failure

The impact on the Siebel CRM deployment is either temporary or long term. For example, a temporary networking interruption, or a quick database server reboot, would result in a temporary disruption in service. A long-term interruption might occur when there is database corruption or a major server malfunction.

In general, user sessions are lost when there is a Siebel database service interruption. Users must log in to the application again. Application Object Manager sessions continue to try to connect to the database. After the database is running (assuming the connection retry count has not been exceeded), the connection succeeds. Users might not notice that there was an outage, unless they were currently working at the time of the database failure. In this case, users get database error messages.

If the interruption is temporary, then the interactive server components and most of the batch-mode server components try to reconnect with the Siebel database. If the interruption is long-term, then the Siebel CRM deployment must be shut down and restarted once the database service is restored.

> **Note:** It is strongly recommended not to collocate a Siebel Server or other components with the Siebel database on the same server hardware. Such a deployment can affect performance of either component, and might in some cases lead to Siebel Server failure or extend interruptions of availability. For example, where a Siebel Server and the Siebel database are collocated, if the server computer is rebooted, then the Siebel database instance might not be up by the time this Siebel Server is up, so a connection could not be made. Siebel Server failure could result. However, if the Siebel Server and the Siebel database are on separate servers, as recommended, then the Siebel database is unaffected by rebooting the server computer where Siebel Server is installed, and is available as soon as the Siebel Server is ready to connect.

Impact of Service Failures

The following table summarizes the impact of failure of services in the Siebel CRM deployment. The table includes information about specific services that were not already covered. This topic is part of *How Service Failures Affect the Siebel CRM Deployment*.

Service Failed	Affected Component	Impact
Siebel Gateway	Siebel Server components	You cannot start or add any new components. Users can continue to log in and out of Siebel applications. Existing user sessions are not interrupted. Server requests continue to be processed successfully, with some exceptions, as follows.
	Server administration functions	Unavailable.

Service Failed	Affected Component	Impact
	Siebel Product Configuration Object Manager	You can still launch product configuration sessions, as long as the connection information has been cached. By default, the connection information is cached when the first connection is made.
	Siebel Gateway registry	This registry maintains server configuration information for the Siebel Enterprise Server. If this database is corrupted or lost, then you must reinstall all of the Siebel Enterprise Server software. You can minimize risks associated with service failures by configuring clustering for Siebel Gateway using Siebel Management Console. For more information, see *Defining High Availability Policies*.
Siebel Server	Application Object Manager components	The Siebel application is unavailable. Siebel Connection Broker (SCBroker) failure: You cannot create new user sessions. If the SISNAPI connection between the Siebel Application Interface and the Object Manager fails, then the Siebel Application Interface retries the connection. If, after a certain number of attempts, the connection is still not available, then the connection completely fails and the user gets an error message. Existing user sessions are unaffected by SCBroker failures.
	EAI	Interface to external application unavailable.
	Batch components	Loss of functionality (components such as Assignment Manager or Workflow unable to process server requests).
Siebel File System	Attachments	Unavailable.
	Correspondence	Unavailable.
	Shared user preference files	Unavailable.
	Docking transaction files from Siebel EIM	Unavailable.
	Siebel Email Response	Unable to process inbound messages. Unable to send outbound messages with attachments.
File System Manager (alias FSMSrvr)	Components that access FSMSrvr	Current requests fail.
	Attachments	Unavailable.
Siebel Application Interface	Siebel Web Clients accessing Application Object Managers	The Siebel application is unavailable to Siebel Web Clients. Siebel Mobile Web Clients are unaffected.

Service Failed	Affected Component	Impact
	EAI inbound HTTP adapter	Unavailable.
Siebel database	Client access, background tasks, batch tasks	Unable to access Siebel CRM applications. Siebel Servers cannot function. Only the Siebel Mobile Web Client is not immediately affected by a Siebel database failure.
	Batch and interactive components	Unavailable.

Specific Failures and Associated Impact

The table in this topic describes potential failure scenarios when running a Siebel CRM deployment, and summarizes benchmark test results and the associated impact on the tested Siebel environment.

This topic is part of *How Service Failures Affect the Siebel CRM Deployment*.

Note: All of the tests were conducted in a test lab. The actual results for production environments might differ due to the level of complexity of the production environment.

The test environment included the following:

- Multiple Application Object Manager servers were deployed with load balancing.
- Multiple batch component application servers were deployed. Request distribution was provided by Server Request Broker (SRBroker) and Server Request Processor (SRProc).
- Siebel Product Configuration Object Manager servers were used and load balanced by the Siebel Product Configurator-provided load balancing scheme.
- A clustered pair of database servers was used.
- A clustered Siebel Gateway was also deployed (using third-party clustering in this case).

Component Tested	Failure Scenario	Observed Behavior
Siebel database	Observe the system behavior while driving server CPU load to 100%.	• Significant response time impact. • No failures were observed.
Application Object Manager (eChannel)	Observe the system behavior while driving server CPU load to 100%.	• Minor response time impact. • No failures were observed.
Siebel Application Interface	Observe the system behavior while driving server CPU load to 100%.	• Negligible response time impact. • No failures were observed.
Workflow Server	Observe the system behavior while driving server CPU load to 100%.	• Negligible response time impact. • No failures were observed.

Component Tested	Failure Scenario	Observed Behavior
Application Object Manager (eChannel)	Observe the system behavior while server memory consumption is 100%.	• Significant response time impact. • Increased CPU usage and context switching were observed. • A few login failures were seen when attempting to log in additional users.
Workflow Server	Observe the system behavior while server memory consumption is 100%.	• Major response time impact. • Increased CPU usage and context switching were observed. • A few login failures were seen when attempting to log in additional users.
Application Object Manager (eChannel)	Observe the system behavior when all of the available disk space is consumed on the tested server.	• Minor response time impact in some transactions. • Major response time impact when logging in new users. • No failures were observed.
Workflow Server	Observe the system behavior when all of the available disk space is consumed on the tested server.	• Significant response time impact in Workflow transaction response time. • Significant response time impact when additional users logged in. • Negligible increase in CPU and context switching. • No failures were observed.
SCBroker	Simulate a process failure for various task-based server components while the server is handling both synchronous and asynchronous server requests. Also note system recovery after bringing the process back up.	• SCBroker auto-restarts upon receiving an SEGV signal. • No failures were observed. • A new SCBroker was started when an SEGV signal was received.
SRBroker	Simulate a process failure for various task-based server components while the server is handling both synchronous and asynchronous server requests. Also note system recovery after bringing the process back up.	• SRBroker does not auto-restart upon receiving an SEGV signal. • When eScripting invokes a workflow, users get a `no server connect string` error message. • Failures were seen for the preceding step.
WfProcMgr (Workflow Process Manager)	Simulate a process failure for various task based server components while the server is handling both synchronous and asynchronous server requests. Also note system recovery after bringing the process back up.	• Shutdown WfProcMgr on one server caused a few failures initially and then stabilized with no further failures. • CPU and memory activity increased on the server still running WfProcMgr. • When the other WfProcMgr is shut down many failures resulted. • Brought up one WfProcMgr and no more failures were observed.

Component Tested	Failure Scenario	Observed Behavior
Siebel Gateway	Simulate the failure of the Siebel Gateway.	• Unable to connect to srvrmgr but the transactions were passing. • When adding 100 more users, still unable to connect to srvrmgr but no errors were observed. • When the Siebel Gateway was restarted, still unable to connect to the Gateway.
Application Object Manager	Consume all available tasks on an Application Object Manager and observe the result.	• Application Object Managers fail over to another Object Manager as expected when MaxTasks is reached. • When all of the Application Object Managers are out of tasks, the user receives a **`server busy`** error message. • When some users log out, new users can connect to servers again.
Application Object Manager	Simulate resource leaks while server recycling is enabled, and verify how process recycling works under load.	• New Application Object Manager gets created when MemoryLimit is hit. • Old Application Object Manager remains instantiated for a period of time (even when no more users are running on it), but eventually the old Object Manager is recycled. • When MemoryLimitPercent is reached, then the whole component restarts. All traffic went to the other server.
Application Object Manager	Simulate a thread or process that is not responding.	• Users can still log in to the Application Object Manager with the non-responsive process. • After simulating a non-responsive process, 100 extra users were added. After that, stopping the non-responsive process caused about 40 running users to fail (users on that Application Object Manager). The following can be inferred: o An Application Object Manager with a non-responsive process still receives new connections. o You cannot safely stop a non-responsive process unless you set the component group offline and shut down the whole component group or Siebel Server.

About High Availability Deployment Options

High availability means that a user can access key system services even when the underlying hardware or software for those services fails. For example, if a user synchronization session was interrupted by a failure of the server computer to which it was connected, then the user can reconnect to a Siebel Remote Server and restart the synchronization process without any data loss.

To achieve high availability, the overall system must automatically replace lost services and distribute loads among services to assure acceptable response times. When a lost service cannot be replaced automatically, this is called a single point of failure. High availability planning and deployment are designed to eliminate these single points of failure.

In a Siebel CRM deployment, a service (for the purposes of this discussion) is one of the following:

- Siebel Gateway (service and registry)
- Siebel Server
- Siebel database
- Siebel File System
- Siebel Application Interface

To eliminate single points of failure, some form of redundancy is required. Clustered servers are an example. When one service fails, other resources are available to take over for the failed service. To be successful, this process must be:

- Automatic: No operator intervention is necessary
- Transparent: Users do not have to change anything for the services that have failover protection

There are cases where full, automatic failover might not be possible. For example, results of the failure might have to be manually cleaned up. This guide does not cover all of the possible cases, and customers are advised to review environment-specific requirements before finalizing high availability planning.

The options available for high availability deployment consist of the following techniques:

- Scalable services (load balancing)
- Resilient processing (distributed services)
- Server clusters

For more information about available deployment options, see *Recommended High Availability Techniques for Specific Services* and *Recommendations for High Availability Deployments*.

Scalable Services (Load Balancing)

Load balancing distributes workload across multiple servers. Each server runs an instance of the service that you want to load-balance. Load balancing also provides failover. If one server fails, then requests are automatically routed to the remaining servers.

Application Object Managers are the server components for which load balancing is most frequently provided. Distributing workload across Application Object Managers indirectly distributes workload across the other server components that Application Object Managers call, in a form of indirect load balancing.

You can also configure load balancing for Siebel Application Interface, as described in *Recommendations for High Availability Deployments*.

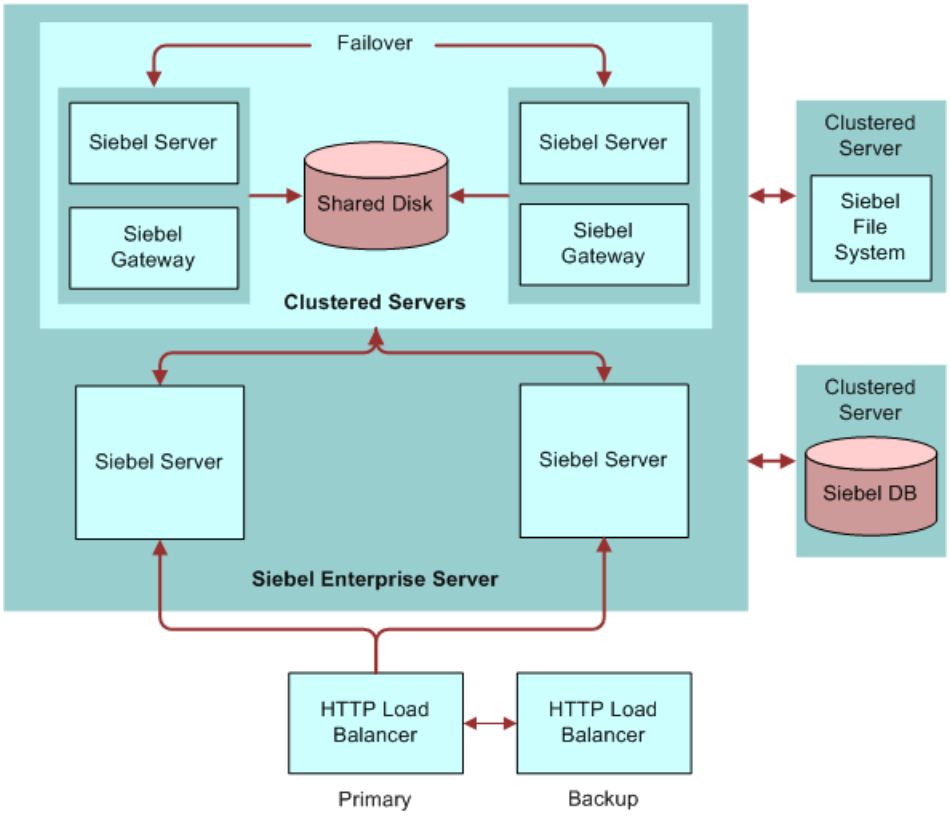
Resilient Processing (Distributed Services)

Resilient processing, also called distributed services, is used for tasks initiated by the Siebel Server. (Load balancing is used for tasks initiated by users.) Multiple instances of a component run on the same Siebel Server, or the same component can run on multiple Siebel Servers. If one instance of the component fails, then another instance on the same server or on a different server takes over processing subsequent requests. For more information, see *About Resilient Processing*.

Server Clusters

Server clusters consist of two or more physical servers linked together so that, if one server fails, then resources such as physical disks, network addresses, and applications can be switched over to the other server. Server clusters can provide resilience when a particular Siebel operation can only take place on one server, either because of the type of process (such as Siebel Gateway or Siebel Remote) or because of hardware constraints.

The following figure illustrates an example of server load balancing and server clustering in a Siebel Enterprise Server.



Note: Siebel CRM supports an optional native clustering feature for Siebel Gateway to provide high availability benefits to Siebel CRM customers. This feature works at the software level and is the preferred and recommended approach for clustering the Siebel Gateway. For more information, see *Defining High Availability Policies*.

Recommended High Availability Techniques for Specific Services

The three supported high availability techniques are server clustering, load balancing, and resilient processing. The following table lists the recommended high availability technique for specific Siebel Enterprise deployment services:

- **Preferred.** Indicates that more than one high availability technique is supported for this function, but this is the preferred technique to use wherever possible.

- **Supported.** Indicates a high availability technique is supported for this function. You can use this technique if local conditions prevent using the preferred technique.
- **N/A.** The high availability technique in this column is not applicable for this component.

Component	Clustering	Load Balancing	Resilient Processing
Siebel Gateway service	Preferred	N/A	N/A
Siebel Gateway registry	Preferred	N/A	N/A
Application Object Managers	Supported	Preferred	N/A
Communications Session Manager	Supported	N/A	Preferred
Siebel Product Configurator	Supported	Preferred. Uses its own load balancing method.	N/A
Siebel Document Server	Supported	N/A	Preferred
Siebel Pricer	Supported	N/A	Preferred
Siebel EAI (adapters and connectors)	Supported	Preferred, whenever possible	Supported
EAI Object Manager	Supported	Preferred	N/A
Field Service (non-Application Object Manager components such as Appointment Booking System and Scheduling)	Supported	N/A	Preferred
File System Manager	Supported	N/A	Preferred
Interactive Assignment	Supported	N/A	Preferred
MQ Series Receiver	Preferred	N/A	N/A
Replication Agent	Preferred	N/A	N/A
Siebel File System	Supported	N/A	N/A
Siebel Marketing	Supported	N/A	Preferred
Siebel Remote component group (alias Remote)	Preferred	N/A	N/A

Component	Clustering	Load Balancing	Resilient Processing
Disconnected Mobile Synchronization component group (alias MobileSync)			
Workflow Monitor Agent	Preferred	N/A	N/A
Workflow Process Manager	Supported	N/A	Preferred

Additional Information

Siebel Document Server clustering is supported where Microsoft Office has been installed on all clustered nodes. This approach can be particularly helpful in smaller deployments.

There are many different types of Siebel EAI deployments and providing a single, standardized recommendation is not practical. For help determining the best approach for your deployment, contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services.

Recommendations for High Availability Deployments

Use the recommendations in this topic as a starting point for planning a high availability infrastructure.

This topic includes the following information:

- *Profile 1: Uninterrupted Global Deployment*
- *Profile 2: Large Domestic Deployment*
- *Profile 3: Limited Resources Deployment*
- *Profile 4: Application Integration Deployment*

Profile 1: Uninterrupted Global Deployment

This deployment has several hundred to tens of thousands of users worldwide requiring the Siebel application to be available 24 hours a day, seven days a week.

This topic is part of *Recommendations for High Availability Deployments*.

- **Siebel Gateway.** You can optionally configure clustering for Siebel Gateway using Siebel Management Console. For more information, see *Defining High Availability Policies*. Alternatively, you can use a dedicated, clustered server pair for this component, or include it with Siebel Servers in an existing cluster. Sharing the clustered servers has a minimal performance impact.
- **Siebel File System.** Consider deploying fault-tolerant and resilient file systems to host the files. Clustering the server that hosts the Siebel File System is also an appropriate strategy. The File System is restricted to one for each Siebel Enterprise Server. Therefore, you cannot use load balancing.
- **Siebel Application Interface.** Application containers for Siebel Application Interface instances on multiple nodes can be load balanced using Apache HTTP Server (httpd) and Apache Tomcat Connector (mod_jk). For more information, see *Components Involved in Service Failures*.

- **Siebel Servers hosting an Application Object Manager.** Consider Application Object Manager or server failure when doing capacity planning. For example, if each Siebel Server can handle 500 users, and you typically have 1500 concurrent users, then consider providing four Siebel Servers to handle this load. If one server fails, then the other three servers can still support user loads.

  The Siebel Product Configuration Object Manager is an exception, because it includes an internal load balancing mechanism.
- **Siebel Servers hosting other types of components.** Enable batch components on multiple Siebel Servers. Server Request Broker routes requests to these components, thus providing resilient processing for batch requests.

  Some components can be hosted on only one Siebel Server, for example Siebel Remote. If user loads permit, then you set up high availability as follows:

  - For the Application Object Manager and related components, use load balancing.
  - For the components that can be installed on only one server, use server clustering.
- **Siebel database.** Deploy the high availability clustered services provided or supported by the vendor of your RDBMS.

  To guarantee data availability and integrity, use data replication techniques such as mirroring and disk arrays to keep the backup instance of the database in sync with the primary instance.

  Also consider fault-tolerant file systems to host database files.

Profile 2: Large Domestic Deployment

This deployment has several hundred to several thousand users in an Enterprise deployment that is operational during standard business hours only.

This topic is part of *Recommendations for High Availability Deployments*.

- **Siebel Gateway.** You can optionally configure clustering for Siebel Gateway using Siebel Management Console. For more information, see *Defining High Availability Policies*. Alternatively, you can use a dedicated, clustered server pair for this component, or include it with Siebel Servers in an existing cluster. Sharing the clustered servers has a minimal performance impact.
- **Siebel File System.** Deploy a clustering technology that has been certified by Oracle. At a minimum, use a RAID 5 disk array for your file system. In addition, make regular backups of your data.
- **Siebel Application Interface.** Application containers for Siebel Application Interface instances on multiple nodes can be load balanced using Apache HTTP Server (httpd) and Apache Tomcat Connector (mod_jk). For more information, see *Components Involved in Service Failures*.
- **Siebel Servers hosting an Application Object Manager.** Consider Application Object Manager or server failure when doing capacity planning. For example, if each Siebel Server can handle 500 users, and you typically have 1500 concurrent users, then consider providing four Siebel Servers to handle this load. If one server fails, then the other three servers can still support user loads.
- **Siebel Servers hosting other types of components.** Same as Profile 1.
- **Siebel database.** Deploy a clustering solution supported by your RDBMS vendor. To guarantee data availability and integrity, use data replication techniques such as mirroring and the disk arrays to keep the backup instance of the database in sync with the primary instance.

Profile 3: Limited Resources Deployment

This deployment has 500 users or less and operates during standard business hours with limited hardware resources.

To establish high availability, consider putting the Siebel CRM deployment in a two-system cluster, clustering Siebel Gateway, Siebel database, and Siebel File System. You can optionally configure clustering for Siebel Gateway using Siebel Management Console. For more information, see *Defining High Availability Policies*.

This topic is part of *Recommendations for High Availability Deployments*.

Profile 4: Application Integration Deployment

This deployment uses third-party application servers to access the Siebel application. There are multiple integration points between Siebel applications and other, third-party applications. This profile might use Siebel EAI extensively.

There are no unique high availability requirements for this profile. See the previous discussions of the other profiles.

Make sure that the third-party applications are highly available by reviewing the specifications published by those vendors.

This topic is part of *Recommendations for High Availability Deployments*.

About Resilient Processing

Resilient processing, also called distributed services, distributes server requests to multiple instances of batch-mode server components. The server requests for these components are typically message-based, so any instance of the component can process the request. If one instance of a component fails, then another instance can perform the task, thus providing resiliency. Multiple instances of the components can run on the same Siebel Server or on several Siebel Servers.

Load balancing is about distributing workloads. Resilient processing is about providing redundancy. Resiliency also provides round-robin distribution of workloads to multiple instances of server components.

Resilient processing makes more efficient use of hardware resources than server clustering. In addition, resilient processing does not require third-party clustering software. Where possible, use resilient processing instead of server clustering, or use them in combination.

Resilient processing is the preferred method for providing high availability for the following server components:

- Communications Session Manager
- Siebel Document Server
- Siebel Pricer
- Siebel Field Service
- File System Manager
- Interactive Assignment
- Siebel Marketing

- Workflow Process Manager

Resilient processing uses two server components:

- **Server Request Broker.** This component handles all server requests. See *About the Server Request Broker*.
- **Server Request Processor.** This component handles asynchronous server requests. See *About the Server Request Processor*.

5 Server Clustering Planning

Server Clustering Planning

This chapter provides information about planning server clustering for your Siebel CRM deployment. It includes the following topics:

- *About Server Clustering*
- *Where to Use Server Clustering*
- *Recommendations for Server Clustering*

About Server Clustering

A server cluster is a group of two or more servers that are configured so that, if one server fails, another server can take over application processing. The servers in a cluster are called nodes. Typically, these servers store data on a common disk or disk array.

Clustering software monitors the active nodes in a server cluster. When a node fails, the clustering software manages the transition of the failed server's workload to the secondary node.

When a clustered Siebel Server fails, all of the applications and services on the server stop. Application users must reconnect and log in to the server that takes over. For example, if the Siebel Server that failed was hosting Siebel Communications Server, then the communications toolbar is disabled, and users must reconnect and log in to the new server.

Cluster vendors can validate their third-party server cluster products to provide server clustering for deployments of Siebel CRM applications. For validation assistance, contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services. For recommendations and help on the use of cluster products with Siebel CRM, customers are advised to contact the cluster vendor of their choice.

Siebel CRM supports an optional native clustering feature for Siebel Gateway, to provide high availability benefits to Siebel CRM customers. This feature works at the software level and is the preferred and recommended approach for clustering the Siebel Gateway. For more information, see *Defining High Availability Policies*.

This topic contains the following information:

- *Active-Passive Configuration*
- *Active-Active Configuration*

Active-Passive Configuration

An active-passive server cluster contains a minimum of two servers. One server actively runs applications and services. The other is idle. If the active server fails, then its workload is switched to the idle server, which then takes over application processing.

Because the standby server is idle, active-passive server clusters require additional hardware without providing additional active capacity. The benefit of active-passive clusters is that, after a failover, the same level of hardware

resources is available for each application, thereby eliminating any performance impact on users. This benefit is particularly important for performance-critical areas such as the database. The most common use of active-passive clusters is for database servers.

This topic is part of *About Server Clustering*.

Active-Active Configuration

An active-active server cluster contains a minimum of two servers. Both servers actively run applications and services. Each server might host different applications or might host instances of the same application. If one server fails, then its processing load is transferred to the other server.

Active-active configuration is the most common server clustering strategy for servers other than the database server.

> **Note:** Configuring the Siebel database (database server) and a Siebel Server to fail over to each other is supported, but not recommended.

This topic is part of *About Server Clustering*.

Potential Port Conflicts

Some Siebel Server components, such as Siebel Connection Broker (SCBroker), Siebel Gateway, and Siebel Remote Synchronization Manager, listen on a configurable static port. When these components run in an active-active cluster, you must plan your port usage so there is no port conflict after failover.

For example, an active-active server cluster contains two computers, each running a Siebel Server. If one computer fails, then the other computer hosts two Siebel Servers. Siebel Servers include a number of services, such as Siebel Connection Broker, that use a dedicated port. If this port number was the same on both computers, then a port conflict occurs after failover.

Capacity Planning

Active-active clusters use all of the server platforms continuously. Consequently, they take better advantage of computing resources than active-passive clusters. When doing capacity planning, make sure that clustered servers have sufficient capacity to handle a failover. Because failovers are usually infrequent and normally last only a short time, some performance degradation is often acceptable.

An active-active server cluster contains a minimum of two servers. Both servers actively run applications and services. Each server might host different applications or might host instances of the same application. If one server fails, then its processing load is transferred to the other server.

Active-active configuration is the most common server clustering strategy for servers other than the database server.

> **Note:** Configuring the Siebel database (database server) and a Siebel Server to fail over to each other is supported, but not recommended.

Where to Use Server Clustering

Siebel CRM supports server clustering for the following parts of a Siebel CRM deployment:

- Siebel Gateway

  Alternatively, you can configure clustering for Siebel Gateway using Siebel Management Console. For more information, see *Defining High Availability Policies*.
- Siebel Servers

  Individual server components can be clustered. Some Application Object Manager components do not support or require clustering.
- Siebel File System
- Siebel database

  Subject to limitations of third-party RDBMS software.
- Siebel Application Interface

In addition, server clustering is the preferred method for providing high availability for the following Siebel Server components:

- Workflow Monitor Agent
- Siebel Remote

  The DockString parameter in the Siebel Mobile Web Client configuration file must reference the virtual server name for remote synchronization to work after failover.
- Replication Agent
- MQSeries Server Receiver

Server Clustering for Some Components Is Not Supported

Siebel CRM does not support server clustering for the following components:

- LDAP directory server. The vendor might provide built-in replication.
- Siebel Document Server (Microsoft server-side integration).
- CTI hardware or switch.

Recommendations for Server Clustering

The following recommendations help promote failover protection for your Siebel CRM deployment. However, these practices are neither exhaustive nor all-inclusive.

- If you have multiple Siebel Servers running that are not clustered, then load balance these servers.
- Make clustering the Siebel database a high priority because it is a single point of failure. When clustering the Siebel database, have it already installed and running. Cluster the Siebel database server first.

- Install and configure clustering software on each node to detect failure of that node and to recover and manage all of the servers as a single system.
- Make sure all of the hardware used is certified for server clustering by the hardware vendor.
- If you operate the Siebel Gateway and Siebel Servers as part of a cluster, then you must install and configure the Siebel Gateway and the Siebel Server individually as separate cluster services.

  Siebel CRM supports an optional native clustering feature for Siebel Gateway, to provide high availability benefits to Siebel CRM customers. This feature works at the software level and is the preferred and recommended approach for clustering the Siebel Gateway. For more information, see *Defining High Availability Policies*.
- On the copy that you made of the cluster deployment worksheet (which is located in *Siebel Installation Guide*), or your custom worksheet, fill out the section related to server clustering, and refer to it during installation.

6 Data Integrity and Capacity Planning

Data Integrity and Capacity Planning

This chapter provides information about planning for database capacity and data integrity in your Siebel CRM deployment. It includes the following topics:

- *Sizing the Database for a Siebel CRM Deployment*
- *Database Table Planning*
- *Database Recovery Planning*
- *Database Physical Device Planning*
- *Database RAID Array Planning*

Sizing the Database for a Siebel CRM Deployment

As with most client-server applications, the overall performance of Siebel CRM applications is largely dependent on the input and output (I/O) performance of the database server. To promote optimal performance for I/O, you must arrange the tables and indexes in the database across available disk devices in a way that evenly distributes the I/O load.

The mechanism for distributing database objects varies by RDBMS, depending on the way storage space is allocated. Most databases can force a given object to be created on a specific disk.

To verify which RDBMS products, versions, and patch levels are supported, see the Certifications tab on My Oracle Support.

In your planning, you must allocate space for multiple purposes, including system storage space, undo or rollback space, temporary table space, and space for logs and system files, as well as space for Siebel data and indexes. If you allocate too little space for your system storage, then you reduce performance. If you allocate too much space, then you waste disk space.

The space the RDBMS needs varies primarily based on the total number and types of users supported, as well as the transaction mix and rate. Consult the RDBMS vendor's documentation for more information about these requirements.

The space required for Siebel data and indexes varies depending on what Siebel CRM functionality you implement and the amount and nature of data supporting that functionality.

Note: The Siebel Servers in a Siebel Enterprise Server can connect to only one database.

To determine the size of the database required for a Siebel CRM deployment

1. Determine the total number and types of users of Siebel CRM applications (for example, 500 sales representatives and 75 sales managers).

2. Determine the Siebel CRM functionality that you implement and the entities that are required to support them. Usually, the largest entities are as follows:
    - Accounts
    - Activities
    - Contacts
    - Forecasts
    - Opportunities
    - Service requests
3. Estimate the average number of entities for each user (for example, 100 accounts for each sales representative) and calculate an estimated total number of records for each entity for your total user base.
4. Using standard sizing procedures for your specific database, calculate the average record size for each entity and multiply by the total number of records.

    Typically, these entities span multiple physical tables, all of which you must include in the row size calculation in order to determine the estimated data size for the largest entities.
5. Add additional space for the storage of other Siebel data. A rough guideline for this additional amount would be half the storage required for these key entities.
    - Indexes typically require approximately the same amount of space as data.
    - Factor growth rates into your total size calculation.
    - Factor a margin of error into your total size calculation.

Database Table Planning

In most enterprise deployments, the following Siebel tables, and their corresponding indexes, are the most commonly used tables or are very large.

For example, the tables S_EVT_ACT, S_CONTACT, and S_ORG_EXT are large in all enterprise-level deployments of Siebel CRM. Separate these tables and indexes across devices.

As a general rule, put indexes in a different table space and, if possible, on different physical devices from the tables on which they are created.

For Siebel table spaces on an IBM DB2 database, use database-managed table spaces (DMS) rather than system-managed table spaces (SMS).

- S_ACCNT_CHRCTR
- S_ACCNT_CO_MSTR
- S_ACCNT_POSTN
- S_ADDR_ORG
- S_ADDR_PER
- S_ASSET
- S_CALL_LST_CON
- S_CON_CHRCTR

- S_CON_TERR
- S_CRSE_TSTRUN
- S_CRSE_TSTRUN_A
- S_CS_RUN
- S_CS_RUN_ANSWR
- S_CTLGCAT_PATH
- S_CYC_CNT_ASSET
- S_DNB_CON_MRC
- S_DNB_ORG
- S_DNB_ORG_SIC
- S_DNB_UPDATE
- S_DOCK_INIT_ITEM
- S_DOCK_TXN_LOG
- S_DOCK_TXN_LOGT
- S_DOCK_TXN_SET
- S_DOCK_TXN_SETT
- S_ESCL_ACTN_REQ
- S_ESCL_LOG
- S_ESCL_REQ
- S_EVT_ACT
- S_EXP_ITEM
- S_EXP_RPT
- S_EXP_RPT_APPR
- S_IC_CALC
- S_IC_CALC_IT
- S_IC_CMPNT_EARN
- S_IC_TXN
- S_IC_TXN_IT
- S_IC_TXN_POSTN
- S_INVC_ITM_DTL
- S_INVLOC_ROLLUP
- S_INVOICE
- S_INVOICE_ITEM
- S_INV_LGR_ENTRY
- S_OPTY_POSTN
- S_OPTY_PROD
- S_OPTY_TERR

- S_ORG_EXT
- S_ORG_TERR
- S_PARTY
- S_PARTY_PER
- S_PARTY_REL
- S_PARTY_RPT_REL
- S_POSTN_CON
- S_PROC_REQ
- S_PROD_BASELINE
- S_PROD_CONSUME
- S_PROD_SHIPMENT
- S_PROD_TARGET
- S_QUOTE_ITEM
- S_SRM_REPLY
- S_SRM_REQUEST
- S_SRM_REQ_PARAM
- S_SRV_REQ

Database Recovery Planning

Follow the RDBMS vendor's recommendations on configuring the database for recovery in case of data corruption, hardware failure, or disaster.

Oracle Database Recovery Planning

Many companies today use RAID storage systems that make Oracle Database online redo log mirroring unnecessary.

If your organization does not use RAID storage systems, then mirror the redo log, at a minimum, because the redo log is essential when a database goes through failure recovery.

Also, when redo logs are mirrored at the RAID storage system level (usually RAID 1 or RAID 0+1), there is usually no need to mirror them at the Oracle Database level, since the RAID controller assures that these volumes can always be recovered. Mirroring at the RAID level usually improves database performance (especially beneficial for read operation).

If you have the resources, then mirror the Oracle Database control files as well. Otherwise, you can put the Oracle Database control files into a RAID 5 device, as it is not heavily accessed and disk performance is not a concern. The information it records, though, is very critical for Oracle Database. Any updates to the control file, such as the current System Change Number (SCN) or transaction tables, ripple across all of the members of the control file specification.

IBM DB2 Recovery Planning

Mirror the transaction log to guarantee database recovery if a single device fails. You must mirror the instance home directory, if resources are available. Hardware or operating system mirroring generally provides the best performance.

Database Physical Device Planning

To make sure that your database performs well, create at least one container for each available logical or physical disk device. You can use table spaces to place objects on multiple physical containers to promote parallel input and output (I/O). Spreading the data and index information across several containers (physical devices) can improve the performance of queries.

IBM DB2 Physical Device Planning

Locate data and log devices on different disk spindles to reduce contention between random and serial I/O. For IBM DB2, locate these devices on different disk spindles to minimize I/O contention. When this approach is not possible, spread devices containing database objects that are often used together across different spindles. These objects include tables, their indexes, and commonly joined tables.

If you are using a high performance disk subsystem, then you might choose a different physical device layout. Consult your DBA and the disk subsystem vendor for the optimal setup.

Physical Device Planning for UNIX Deployments

For UNIX database servers, locate all of the containers on raw UNIX disk partitions, except the containers used for `LONG VARCHAR` data. Locate containers for `LONG VARCHAR` data on the UNIX file system to take advantage of the operating system's buffering capabilities. To make sure that your database performs well, create one container for each available logical or physical disk device. Locate data and log devices on different disk spindles to reduce contention between random and serial I/O.

Microsoft SQL Server Physical Device Planning

Use filegroups for assigning database objects to one or more files within a filegroup for maximum performance of the Siebel database. When you group objects, you have the ability to distribute a filegroup across multiple disks, thereby causing less resource contention.

If your enterprise does not require very high performance, based on the number of concurrent users, for example, then using RAID devices and Microsoft's default setting might suffice. A database administrator must do the necessary sizing calculations to assess the performance requirements during the planning process.

Database RAID Array Planning

A database RAID array (redundant array of independent drives) can provide large amounts of input and output (I/O) throughput and capacity, while appearing to the operating system and RDBMS as a single large disk (or multiple disks, as desired, for manageability). The use of RAIDs can greatly simplify the database layout process by providing an abstraction layer above the physical disks, while promoting high performance.

Performance of the RAID feature provided by the operating system might not be satisfactory. To obtain the best RAID performance, use the RAID support provided by your RAID vendor.

If a RAID Array Is Not Used

If a RAID device is not in use, even if space is at a premium, then you must separate indexes with names ending in _P1 from the tables on which they are created. These tables are heavily used in joins.

If you make frequent use of Siebel Enterprise Integration Manager (EIM), then you might want to put the EIM tables and indexes (names starting with EIM_) on different devices from the Siebel base tables. Both tables are accessed simultaneously during Siebel EIM operations.

Microsoft SQL Server RAID Array Planning

The following table describes a sample disk layout for a server dedicated to Microsoft SQL Server, where the database uses a single filegroup residing on a disk array. The use of a single RAID array for the database devices provides satisfactory performance in many cases without the administrative overhead of using individual filegroups.

Disk	Objects	Comments
Single mirrored	Windows OS	N/A
Single disk	Windows pagefile	Segregate for maximum performance.
Single mirrored	SQL Server logfile	Segregate sequential I/O for database performance.
3 to 5 disks (minimum) in a RAID configuration	Siebel database data and indexes	Add as many spindles as required for performance and storage capacity.

If your enterprise requires the highest performance standards, then place heavily used tables and their corresponding indexes, such as those listed under *Sizing the Database for a Siebel CRM Deployment*, in a specific SQL Server filegroup within your database. By creating a filegroup on a specific disk or on multiple disks, you can control where tables and indexes in your database are physically located. For more information, see *Database Physical Device Planning*.

When separating database objects into filegroups, you can avoid complex calculations by using Microsoft's recommended RAID disk layouts.

Your choice to use RAID devices or multiple filegroups to distribute database objects depends solely on how great your performance needs are. It is recommended that you work with your hardware vendor to determine the optimal RAID configuration for your specific requirements.

7 Application-Level Deployment Planning

Application-Level Deployment Planning

Some Siebel applications or product modules can be deployed in multiple ways. This chapter provides an overview of deployment options for these applications. Use this chapter to help make decisions about how to deploy applications across Siebel Servers. This chapter includes the following topics:

Session Communications

- *Session Communications Server Components*
- *Session Communications Performance Factors*
- *Session Communications Deployment Planning*

Siebel Email Response

- *Siebel Email Response Server Components*
- *Siebel Email Response Performance Factors*
- *Siebel Email Response Deployment Planning*

Siebel Product Configurator

- *Siebel Product Configurator Server Components*
- *Siebel Product Configurator Architecture*
- *Siebel Product Configurator Performance Factors*
- *Siebel Product Configurator Deployment Planning*

Siebel Workflow Manager

- *Siebel Workflow Deployment Planning*

Siebel Remote and Batch Job Processing

- *Planning Batch Processing When Using Siebel Remote*

For more information about application deployment planning, see *Siebel Performance Tuning Guide* , *Siebel Installation Guide* , *Siebel System Administration Guide* , and other relevant guides identified in this chapter.

Session Communications Server Components

Session communications refers to using Siebel Communications Server components to enable contact center agents or other users to handle interactive communications work items. For example, Siebel CTI and Siebel Chat support this capability, enabling agents to handle voice calls or chat interactions using the communications toolbar.

Siebel Communications Server provides an application environment to support several kinds of communications activities for Siebel application users, including session communications (such as voice calls or chat). For more information about Siebel CTI, see *Siebel CTI Administration Guide* . For more information about Siebel Chat, see *Siebel Chat Guide* .

Key Siebel Server Components

Session communications are supported in the Siebel Server environment primarily by the following components:

- **Communications Session Manager (CommSessionMgr).** This server component manages interactive communications work items such as voice calls.
- **Application Object Manager.** This server component, such as Call Center Object Manager, manages application sessions for end users who use the Siebel Web Client, including users who handle communications work items (agents). Interactive communication requests from agents typically go through the Application Object Manager.
- **Server Request Broker (SRBroker).** This server component handles communications between the Application Object Manager and certain other Siebel Server components, including CommSessionMgr.

  For example, when a Siebel CTI agent makes a call through the communications toolbar, the request goes from the Application Object Manager to CommSessionMgr by way of SRBroker.

  SRBroker is used whether CommSessionMgr runs on the same computer as the Application Object Manager, or on a different computer.

Additional Siebel Server Component

You might also be using the following Siebel Server component to manage session communications:

- **Communications Configuration Manager (CommConfigMgr).** Optionally, you might use this server component to cache communications configuration data.

Session Communications Performance Factors

Depending on your deployment, your agents might handle phone calls (Siebel CTI) or work items of other communications channels, or some combination of these. Use the following factors to analyze system performance:

- **Inbound calls processed per hour.** This is the number of inbound calls (or other types of work items) processed per hour (or some other time period) by your communications infrastructure.
- **Outbound calls processed per hour.** This is the number of outbound calls processed per hour (or some other time period) by your communications infrastructure. (For outbound predictive dialer calls, only the calls that are answered and processed by Communications Server are relevant here.)
- **Number of user communications actions per minute (load).** This is the average number of communications-related user actions per minute, and the average think time between such user actions. Communications-related actions typically refers to actions performed using the communications toolbar.

  Longer think times mean less load on the Siebel database and Siebel Server. Think time is an important factor in the overall system load. Approximate actual user usage in your estimations.
- **Number of concurrent communications users (agents).** This is the number of concurrent users of session communications features, typically, contact center agents. This figure is some percentage of the total number of concurrent users on the Application Object Manager.
- **Number of work items.** This represents the average number of inbound and outbound work items for each agent, and how these factors relate to your organization's service goals are also important factors influencing performance. Some agents receive a large number of work items from ACD queues, or initiate a large number

of work items. Supervisors or other users might be defined as agents but might receive only escalated work items, for example.

- **Volume of customer data.** This is the total volume of customer data. Data volume affects how quickly data can be retrieved for various purposes, such as to perform lookups for pop-up windows, route work items, or populate the customer dashboard. In many cases, data volume directly affects agents' response times. Assume a realistic volume of data and tune the database to reflect real-world conditions.

Third-Party Product Considerations

Review information presented in applicable third-party documentation for any requirements that affect your deployment. For example:

- Some CTI middleware software might place limitations on the number of agents that can be served at a single contact center site.
- Integration with ACD queues, predictive dialers, or other modules might affect your configurations, affect network traffic, or have other impacts.
- The capacity of your telephony link (between the ACD switch and the CTI middleware) can affect performance.

Session Communications Deployment Planning

Generally, you run Siebel Communications Server components for session communications, such as CommSessionMgr, on the same Siebel Server computers as those running Application Object Managers. In some cases, however, you must run CommSessionMgr on a different computer than the Application Object Managers. These options are described in detail, as follows. CTI middleware generally runs on servers located at each contact center facility.

Running CommSessionMgr on Application Object Manager Computers

Generally, you run Siebel Communications Server components for session communications on the same Siebel Server computers as those running Application Object Managers. Such a topology allows the Application Object Manager load balancing mechanism to indirectly balance Communications Server load. CommSessionMgr loads are fairly light and do not, in themselves, present a reason to run this component on dedicated computers.

Set the Enable Communication parameter to True for all of the Application Object Managers to which your agents connect. If you use load balancing, then configure all of the Application Object Managers to which requests are distributed in the same way.

Running CommSessionMgr on Dedicated Computers

Sometimes you must run CommSessionMgr on a different computer than the Application Object Manager components.

CommSessionMgr must run on the same computer where the communications driver for your CTI middleware is running. If your driver requires a particular operating system, then you must install Siebel Server and run CommSessionMgr on a computer with that operating system. Communications drivers must be able to run on one of the supported Siebel Server platforms, as described in the Certifications tab on My Oracle Support.

Siebel Email Response Server Components

Siebel Email Response uses Communications Server components to enable contact center agents to read and respond to inbound email messages. For more information about Siebel Email Response, see *Siebel Email Administration Guide* .

Key Siebel Server Components

Siebel Email Response is supported in the Siebel Server environment primarily by the following server components:

- **Communications Inbound Receiver (CommInboundRcvr).** This server component receives inbound work items and queues them for processing by Communications Inbound Processor. Work items might include email messages (for Siebel Email Response).
    - For nonreal-time work items, such as email messages for most deployments of Siebel Email Response, Communications Inbound Receiver queues work items it has received for further processing by Communications Inbound Processor.
    - For real-time work items, such as email messages for some deployments of Siebel Email Response, Communications Inbound Receiver processes work items it has received. Communications Inbound Processor is not used.
- **Communications Inbound Processor (CommInboundProcessor).** This server component processes inbound work items that were queued by Communications Inbound Receiver.
- **Communications Outbound Manager (CommOutboundMgr).** This server component sends outbound email.
- **Siebel File System Manager.** This server component writes to and reads from the Siebel File System. It stores inbound messages prior to processing and stores attachments to inbound and outbound email messages.

Additional Siebel Product Module

In addition to Siebel Email Response, you might be using the following Siebel module:

Siebel Assignment Manager. You can use this module for routing email messages to agents.

Third-Party Email Server

Siebel Email Response works in conjunction with your third-party email server. Review information presented in documentation for your email server for any requirements that affect your deployment. For information about supported email servers, see the Certifications tab on My Oracle Support.

Siebel Email Response Performance Factors

The key factors that influence performance for Siebel Email Response deployments are as follows:

- **Inbound email messages processed per hour.** This is the number of inbound email messages processed per hour (or some other time period) by your communications infrastructure.

Requirements for processing outbound messages are relatively minor and are tied to inbound message volume. However, you must also consider other usage of the CommOutboundMgr component or of the email system. For example, you can configure the Send Email command to send email through CommOutboundMgr.

- **Volume of customer data.** This is the total volume of customer data, including templates or categories, literature items, and so on. Template format (HTML or plain text) is a related factor.

Other factors include the size and complexity of inbound email messages and outbound replies.

Also relevant are user settings in the Outbound Communications section of the User Preferences screen, such as whether a reply contains the original message (Include Original Message in Reply setting), or whether HTML or plain text is an agent's default message format (Default Message Format setting).

Siebel Email Response coverage in this topic focuses on inbound and outbound email processing. In a multichannel environment, session communications performance issues also apply.

Siebel Email Response Deployment Planning

Processing inbound email messages makes more demands on server resources, particularly CPU usage levels, than processing outbound messages.

A single computer must handle processing of inbound messages associated with a single response group.

If inbound message volume warrants it and if multiple server computers are available to run CommInboundRcvr and related components, then consider running CommInboundRcvr on a separate computer (or computers) from other Communications Server components.

Combining processing of messages for multiple email accounts in a single response group can make processing of inbound messages more efficient. However, if message volume is expected to grow, then limiting the number of email accounts processed by each response group gives you more flexibility to distribute processing across multiple servers, avoiding processing bottlenecks.

Siebel Product Configurator Server Components

Siebel Product Configurator allows users to interactively configure customizable products when ordering or when generating a quote. Siebel Product Configurator uses a constraint-based solution engine that resides on the Siebel Server, called the Constraint Engine. This engine evaluates customer choices and generates product configurations that conform to business rules.

For more information about using, installing, administering, and tuning Siebel Product Configurator modules, see:

- *Siebel Product Administration Guide*
- *Siebel Installation Guide*
- *Siebel System Administration Guide*
- *Siebel Performance Tuning Guide*
- Article ID 2112562.1 on My Oracle Support

Remaining topics about Siebel Product Configurator in this chapter include:

- *Siebel Product Configurator Architecture*
- *Siebel Product Configurator Performance Factors*
- *Siebel Product Configurator Deployment Planning*

Siebel Product Configurator Components

Siebel Product Configurator is supported in the Siebel Server environment by the following components:

- **Application Object Manager.** The Siebel Product Configurator solution engine is called by the Application Object Manager, such as Call Center Object Manager (SCCObjMgr_*lang*, such as SCCObjMgr_fra for French) for Siebel Call Center.
- **Siebel Product Configuration Object Manager (alias eProdCfgObjMgr_locale).** This is a special-purpose Object Manager component suitable for some Siebel Product Configurator deployments. For convenience, the component alias is generally referred to in this guide as simply eProdCfgObjMgr. This component contains the Siebel Product Configurator solution engine. It can be deployed on a separate Siebel Server from where Siebel Product Configurator sessions are invoked.

  The Siebel Product Configurator server in this type of deployment is also referred to as a remote or dedicated Siebel Product Configurator, which works in coordination with the invoking Application Object Manager. See later in this topic for locale-related requirements for a remote Siebel Product Configurator component.
- **Siebel File System.** This component stores cached object definitions for customizable product definitions in the `CFGCache` directory on the Siebel File System.

Locale-Related Requirements for Remote Siebel Product Configurator Components

The three-letter extension to the alias of the Siebel Product Configuration Object Manager component (*jpn* in the example of eProdCfgObjMgr_jpn) corresponds to the value for the Locale Code parameter (alias LocaleCode) associated with the invoking Application Object Manager. For a remote Siebel Product Configurator component that you intend to invoke, the name of the component must follow this pattern.

The reason for this requirement is that data passed between the invoking Application Object Manager and the remote Siebel Product Configurator component is in a locale-specific format.

If Locale Code on the invoking Application Object Manager is set to a value that does not correspond to a language supported for Siebel applications, then you must either change the Locale Code on the invoking Application Object Manager or create a new remote Siebel Product Configurator component with the required name.

For example, assume the invoking Application Object Manager is SCCObjMgr_enu, but Locale Code is set to ENG rather than ENU. In this case, you must do one of the following:

- Change Locale Code to ENU in order to work with a remote Siebel Product Configurator component named eProdCfgObjMgr_enu.
- Create a remote Siebel Product Configurator component named eProdCfgObjMgr_eng.

In addition, note that the Language and Locale Code parameter settings for the remote Siebel Product Configurator component must match the parameter settings for the invoking Application Object Manager component.

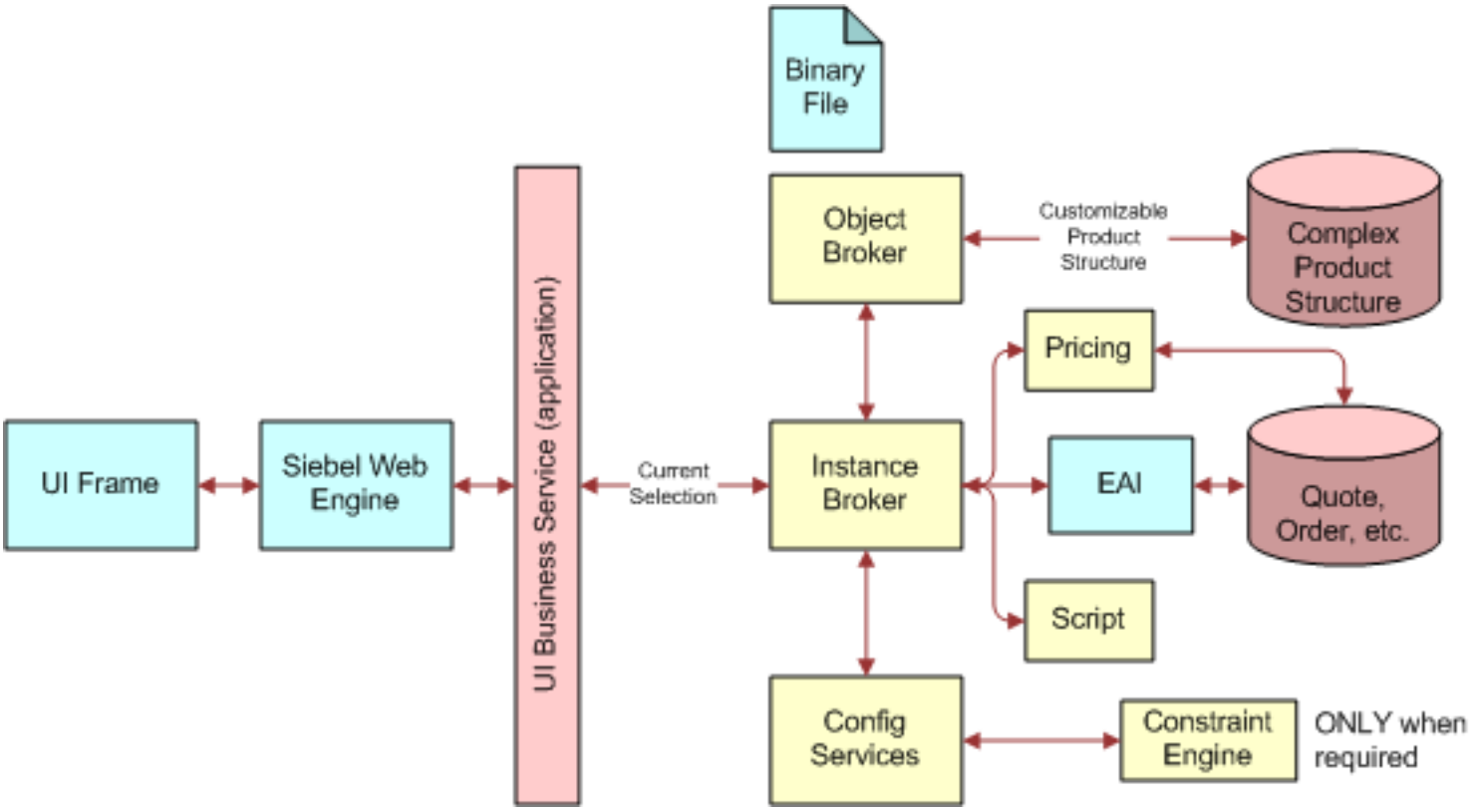
For more information about the Language and Locale Code parameters, see *Siebel Global Deployment Guide* . For more information about creating and configuring server components, see *Siebel System Administration Guide* .

Siebel Product Configurator Architecture

Because product configuration can sometimes be computationally expensive, the configuration infrastructure provides flexible deployment options to suit different business needs. Topics in this section discuss considerations for choosing among different deployment options.

Before the Siebel Product Configurator parameters, and where to set them, are described, a brief overview is presented of the Siebel Product Configurator architecture and of the various services in a Siebel Product Configurator deployment.

The following figure shows detailed Siebel Product Configurator architecture and the interaction of various services with each other during run time.



The important services depicted in the preceding figure are as follows:

- **UI.** The UI business service is a service that the Siebel Product Configurator uses to render the user interface by binding the customizable product structure with the templates and submitting it to the Siebel Web Engine for rendering to the client browser. The UI business service is the way the user interacts with the Siebel Product Configurator. A unique instance of this service is required for each user.
- **Instance Broker.** The Instance Broker is a service that interacts with the UI service and maintains all of the information about the current configuration of the customizable product that the user is configuring. This service interacts with other services in response to user requests during configuration, receives their responses, and serves as backup to the user through the UI service. The Instance Broker is accessed through a proxy service: either the Complex Object Instance Service business service or the Remote Complex Object Instance Service business service.

- **Configurator Object Broker.** The Configurator Object Broker is a service (Cfg Object Broker business service) that extracts the customizable product definition from the database for use by other configuration services.
- **Config Services.** This is a configuration service that consists of factories (defined as follows).
- **Factory.** The factory is a service that represents a translation of the customizable product definition that is retrieved by the Configurator Object Broker into a format a worker (defined as follows) can understand.
- **Constraint Engine or Worker.** The constraint engine, also called a worker, is a service that enforces all of the rules associated with the customizable product. It validates all selections (interactive or batch) as they are made to ensure a valid configuration. A worker of a factory can be shared among different requests originating from the same Application Object Manager process.

For more information about elements of Siebel Product Configurator's internal architecture, including the Instance Broker and the Configurator Object Broker, see *Siebel Product Administration Guide* .

> **Note:** The Oracle Constraint Technology integration for Siebel Product Configurator is no longer available as an alternative way to deploy Siebel Product Configurator. Siebel Constraint Engine is no longer supported for Siebel Product Configurator, as of Siebel CRM 20.12 Update. (This Siebel Constraint Engine is completely different than the Constraint Engine used by the traditional Siebel Product Configurator offering that uses iLog.) For more information, see *Siebel Installation Guide* .

Siebel Product Configurator Performance Factors

For an overview of Siebel Product Configurator server components and other architecture elements, see *Siebel Product Configurator Server Components*.

Siebel Product Configurator performance contexts to consider include response times for:

- **Loading customizable products.** The time elapsed from the moment a user clicks Customize in a quote or order until the Siebel Product Configurator user interface for the customizable product is loaded and displayed to the user.
- **Responding to user selections.** The time elapsed from the moment a user makes a selection until Siebel Product Configurator returns a response, such as an update to the customizable product or a conflict message.

The following performance factors, particularly customizable product size and complexity, are relevant in both of these contexts:

- **Number of concurrent configuration users.** The number of concurrent users who access customizable product definitions. This figure is some percentage of the total number of concurrent users on all of the applicable Application Object Managers.

  Specifically, you would be concerned with the total number of configuration sessions per hour, and the average length of those sessions.
- **Size and complexity of customizable products.** The total size and complexity of each customizable product definition, particularly where multiple hierarchical levels, many constraints, and a complex user interface are defined.

  A major potential performance factor is custom scripting attached to update events on applicable business components, such as Quote, Quote Item, Quote Item Attribute, Order, Order Item, and Order Item Attribute.
- **Number of customizable products.** The number of customizable products accessed by users. It is assumed that each user accesses no more than one customizable product at one time. A given group of concurrent users

might access multiple customizable products, however, each of which must have a separately cached factory. (An existing worker can be shared if one is available. Otherwise, internal performance mechanisms generate a new worker.)

- **Use of SnapShot mode caching.** A feature that caches customizable products in memory, significantly reducing the amount of time required to load customizable products for each new user. SnapShot mode is particularly useful for improving performance when a product line has a small number of large, complex customizable products. For more information about SnapShot mode caching, see *Siebel Product Administration Guide* .

The response time of loading customizable products depends in part on the time taken to extract the customizable product definition from the database, and the time needed to instantiate all of the services required to create the session. At the highest level, the response time for loading a customizable product would be best under the following circumstances:

- The customizable product definition is cached in memory and does not have to be extracted from the database.
- All of the services that are required are already available and do not have to be instantiated.

Caching of objects and services in memory can lead to significant improvement in load time performance for a configuration session. Ideally, everything would be cached, which would give the best possible load time performance. However, that is not possible because the RAM available on the server is limited. For this reason, a caching strategy must be devised for each deployment. To enable administrators to implement the caching strategy that is best suited for their deployment, several switches in the form of server parameters have been provided.

Siebel Product Configurator Deployment Planning

Siebel Product Configurator deployment planning must take into account the considerations described in the following topics:

- *About Deployment Topology for Siebel Product Configurator*
- *About Siebel Product Configurator Caching*
- *Determining Factory and Worker Size*
- *Example of Sizing the Cache with SnapShot Mode*
- *Siebel Product Configurator Deployment Topology Options*
- *Example of Deployment Sizing with a Dedicated Siebel Product Configurator Server*

About Deployment Topology for Siebel Product Configurator

There are two major topology approaches to deploying Siebel Product Configurator:

- Running Siebel Product Configurator in an Application Object Manager component.
- Running Siebel Product Configurator on one or more dedicated Siebel Servers.

For details, see *Siebel Product Configurator Deployment Topology Options*.

This topic is part of *Siebel Product Configurator Deployment Planning*.

Running Siebel Product Configurator in an Application Object Manager Component

You can run Siebel Product Configurator in the Application Object Manager component, such as a language-specific Call Center Object Manager component.

If a small number of concurrent users require configuration sessions, or there is a small number of customizable product definitions, then this deployment option might yield reasonable performance and make the most effective use of your hardware resources.

Running Siebel Product Configurator on Dedicated Computers

You can run Siebel Product Configurator on one or more dedicated Siebel Server computers using a server component called the Siebel Product Configuration Object Manager (eProdCfgObjMgr).

Such server computers are sometimes referred to as remote servers, because they are remote to the computer on which the Application Object Manager is running. In general, this guide uses the term dedicated servers.

If a large number of concurrent users require configuration sessions, or there are a large number of customizable product definitions, then using one or more dedicated Siebel Product Configurator servers might yield the best performance and make the most effective use of your hardware resources.

Possible variations on this deployment strategy include:

- Running one eProdCfgObjMgr component with one Application Object Manager component
- Running multiple eProdCfgObjMgr components with one Application Object Manager component
- Running one eProdCfgObjMgr component with multiple Application Object Manager components
- Running multiple eProdCfgObjMgr components with multiple Application Object Manager components

About Siebel Product Configurator Caching

This topic provides information about Siebel Product Configurator caching.

This topic is part of *Siebel Product Configurator Deployment Planning*.

Configurator Object Broker

The Configurator Object Broker is a service that extracts the customizable product definition from the database for use by other configuration services. To improve performance, the Configurator Object Broker also maintains a cache of objects in the memory to minimize interaction with the database. Normally, the size of the cache is quite small. Different users can share the same object cache.

Factory

The factory is a service that creates a translation of the customizable product definition that is retrieved by the Configurator Object Broker into a format that the worker (described as follows) can understand. Each factory can serve multiple users at run time. Factories are specific to each customizable product, meaning that each customizable product requires its own unique factory. Factories can be cached in memory.

Worker

The worker is shared: in other words, a session only locks a worker during a single configuration request. In between requests, the worker can serve additional configuration sessions.

While the relative number of workers now required to support a user base depends on the time between clicks of particular scenarios and the number of clicks that require worker interaction, the general guideline is that a given worker can now support two to three concurrent users for the same customizable product.

SnapShot Mode

SnapShot mode is a server setting that allows the Siebel Product Configurator to create and execute using cached objects, factories, and workers. If this setting is not chosen, then each user configuration session would be associated with the following:

- Extraction of the customizable product definition and all of the objects associated with it from the database
- Creation of a factory for the customizable product
- Creation of a worker for the user session

If SnapShot mode is chosen, then, depending upon the specified parameter values, objects, factories, and workers would be cached in memory and would be first examined for whether they can be used to initiate a user session. Only if the existing cache cannot support the user session are new objects extracted or factories or workers created.

Considerations About SnapShot Mode

Here are some things to remember about SnapShot mode:

- SnapShot mode determines the upper bound of caching.
- The cache is created as one goes along. The first user request results in the first set of cache data being created. The second user might end up using the same cache because the user wants to configure the same customizable product that the first user configured. Meanwhile, the third user, who wants to configure a different product, creates a new cache, and so on.
- Re-use of cache data occurs only for requests originating from the same Application Object Manager process, for any Siebel Product Configurator deployment.

Determining Factory and Worker Size

This topic provides information about determining factory and worker size for your Siebel Product Configurator deployment.

This topic is part of *Siebel Product Configurator Deployment Planning*.

Factory Size

As a rule, you can assume that the factory size at run time is 75% of the incremental memory used when the customizable product is instantiated.

For example, the factory size equals 75% of (Y minus X), which equals .75 times (40 minus 20), which equals 15 MB.

Worker Size

The worker size varies during run time. Generally, the worker size increases as selections are made. To size the worker, take the maximum memory observed and subtract the factory size from it.

For example, the worker size equals (Z minus X) minus the factory size, which equals (50 minus 20) minus 5, which equals 25 MB.

Object Cache Size

Because the object cache size is normally quite small (for example, 500 KB), you can ignore it in your calculations.

Example of Sizing the Cache with SnapShot Mode

This topic provides an example of sizing the Siebel Product Configurator cache with SnapShot mode.

This topic is part of *Siebel Product Configurator Deployment Planning*.

Assumptions

The requirement is to support 5000 concurrent Siebel Call Center users. Among them, at any time, 100 users use Siebel Product Configurator. This means:

- The enterprise must support 5000 concurrent Call Center users.
- Of these 5000 Call Center users, 100 must be able to use Siebel Product Configurator concurrently.
- There is only one customizable product in the product portfolio.

Sizing

Because all of the caching and services are specific to the Application Object Manager process on a Siebel Server, first you must estimate the size of the Call Center deployment. (The following numbers are used for example only, and are not indicative of Call Center sizing.)

- Assume that you are supporting the 5000 Call Center users on eight application servers (each a Pentium 4 computer with 4 CPUs and 4 GB of memory), with each server handling 625 users.
- Each Siebel Server runs with 25 Application Object Managers, with each Application Object Manager supporting 25 users.

To support cached objects, factories, and workers for all 100 users, the following conclusions can be drawn:

- At least one factory must be cached for every Object Manager process. This means that you must cache 25 factories for each server or one for each Application Object Manager.
- To support all 100 concurrent users to get a cached worker, you must cache, at a minimum, 100 workers across the Enterprise. At the same time, cache at least one worker for each Application Object Manager process. This means that you must cache 25 workers for each server or one for each Application Object Manager.

In the preceding example, the cache size in this case for each Application Object Manager equals the size of the factory cache plus the size of the worker cache. Expressed as a formula, it looks like this: 5 plus 25 equals 30 MB for each Application Object Manager. Therefore, the Siebel Product Configurator cache requires a total of 30 times 25, which equals 750 MB for each server.

The server parameters would be set as follows for each Siebel Server (Application Object Manager):

- eProdCfgSnapshotFlg: True
- eProdCfgNumbOfCachedWorkers: 1

Observations About Sizing

From the preceding sizing exercise, it is clear that the Siebel Product Configurator cache must be actively managed for best performance using appropriate resources. It is extremely important to go through the exercise of sizing the cache. In some cases, the cache requirements might be such that they require additional application servers to fully support all of the users with good response times for load time.

In addition to the preceding calculation, in some situations it is appropriate to set the number of workers according to how much memory is available once enough memory has been allocated to the factory cache and application overhead. The details of this calculation are specific to an individual implementation's average factory size, average worker size, and average Application Object Manager process size. The average Application Object Manager process size depends on the number of Application Object Manager processes, the total memory available, and the maximum process size for the operating system being used.

For additional assistance in this area, contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services.

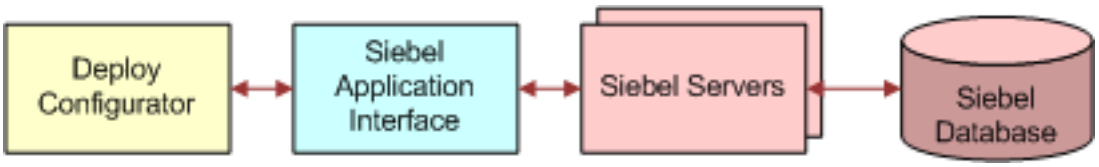
Siebel Product Configurator Deployment Topology Options

Siebel Product Configurator offers the flexibility of different deployment topology options.

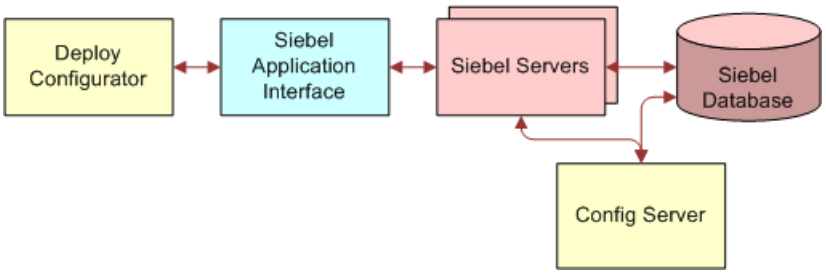
This topic is part of *Siebel Product Configurator Deployment Planning*.

Siebel Product Configurator deployment topology options are as follows:

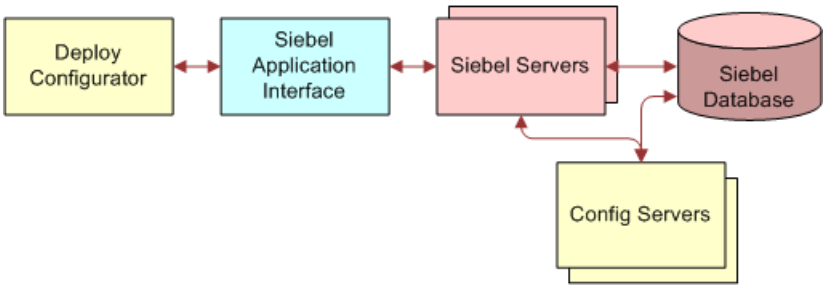
- Deploy Siebel Product Configurator to run on the same computer as the base application server computer, as shown in the following figure.



- Deploy Siebel Product Configurator to run on a different computer than the base application server computer, as shown in the following figure.



- Deploy multiple instances of Siebel Product Configurator on multiple dedicated application server computers, as shown in the following figure.



In many cases, the option of deploying Siebel Product Configurator on a different application server computer might result in a much better use of resources due to pooling effects. Considering the example examined and sized from the preceding topic, the result was a sizing of one factory and one worker to be cached for each Application Object Manager on each server. The implications of this approach across the whole enterprise are as follows:

- The number of Application Object Managers on each server was 25, so each server caches 25 factories and 25 workers.
- Across the whole enterprise of eight application servers, this translates to 25 times 8, which equals 200 factories and 200 workers to be cached.
- In memory terms:
  - 200 times 5, which equals 1000 MB for caching factories, and
  - 200 times 25, which equals 5000 MB for caching workers

  This results in a total memory usage of 6000 MB across the enterprise.

Because the requirement is to support 100 concurrent users, many of these factories and at least 100 of these workers are idling at any time. The large amount of cache in this case is because there is no way to know in advance which Application Object Manager process the user who is configuring is connected to. For this reason, caching must be done across all of the Application Object Managers.

There are other problems with this scenario. For instance, what happens when two users are connected to the same Application Object Manager process and both want to configure? In this case, the second user has to create a worker, causing performance issues for that user. Also, if there are multiple users configuring on the same computer, then the computer might run out of memory.

Consider another scenario. Assume that the enterprise has customizable products that these users might be configuring, with 50 concurrent users configuring each customizable product. However, because there is no way to know in advance which Application Object Manager process these users might be connected to, you would have to cache two factories and two workers for each Application Object Manager, which would be a less-than-satisfactory solution.

Example of Deployment Sizing with a Dedicated Siebel Product Configurator Server

Consider the sizing example for the deployment option of running the Siebel Product Configurator on the same server as the application server (see *Example of Sizing the Cache with SnapShot Mode*). Size it instead for the deployment option of Siebel Product Configurator running on a separate server.

This topic is part of *Siebel Product Configurator Deployment Planning*.

Assumptions

The requirement is to support 5000 concurrent Siebel Call Center users. Among them, at any time, 100 users use the Siebel Product Configurator. This means:

- The enterprise must support 5000 concurrent Call Center users.
- Of these 5000 Call Center users, 100 must be able to use the Siebel Product Configurator concurrently.
- There is only one customizable product in the product portfolio.

Sizing

Because all of the caching and services are specific to the Application Object Manager process on a Siebel Server, first you must estimate the size of the Call Center deployment. (The following numbers used are an example only and not indicative of Call Center sizing.)

- Assume that you are supporting the 5000 Call Center users on seven application servers (each being a Pentium 4 computer with 4 CPUs and 4 GB of memory), with each server handling 720 users.
- Each application server itself is run with 25 Application Object Managers, with each Application Object Manager supporting 25 users.
- Assume that one server has been configured to run the Siebel Product Configurator that supports the 100 users.
- The Siebel Product Configurator server is configured to run with four Application Object Managers, with each Application Object Manager supporting 25 users.

To support cached objects, factories, and workers for all 100 users, the following conclusions can be drawn:

- At least one factory must be cached for every Application Object Manager process. You must cache four factories for the Siebel Product Configurator server or one for each Application Object Manager.
- To support all 100 concurrent users to get a cached worker, you must cache, at a minimum, 100 workers across the Siebel Product Configurator server. This means that you must cache 25 workers for each Application Object Manager on the Siebel Product Configurator server.

In the preceding example, the cache size in this case for each Application Object Manager equals the size of the factory cache plus the size of the worker cache. Expressed as a formula, it looks like this: (5 times 1) plus (25 times 25) equals 630 MB for each Application Object Manager. Therefore, the Siebel Product Configurator cache requires a total of 4 times 630, which equals 2520 MB for each server.

The server parameters would be set as follows for the Siebel Servers running the Application Object Managers and the Siebel Product Configurator servers:

Parameter Name	Value	Description
eProdCfgServer	Name of the Siebel Server running the Siebel Product Configurator.	Set On: Each Siebel Server running the Application Object Manager (see the information that follows about server settings for the dedicated Siebel Product Configurator server deployment mode)
eProdCfgSnapshotFlg	True	Set On: Each Siebel Server running the Application Object Manager and each Siebel Product Configurator server
eProdCfgNumbOfCachedWorkers	25	Set On: Siebel Product Configurator server

This type of deployment across an enterprise with eight servers, one a dedicated server to support Siebel Product Configurator, requires 2520 MB of cache. This figure is much lower than the 6000 MB required for the eight application server deployment option. Choosing this deployment option makes better use of the cache.

Moreover, since the Siebel Product Configurator server is configured to allow only 25 connections to each Application Object Manager, there would never be a case where a user does not find a cached worker to work with. In a scenario with multiple customizable products, this deployment would be much more efficient in terms of memory usage.

Server Settings for Dedicated Siebel Product Configurator Server Deployment Mode

The following table shows server settings for dedicated (remote) Siebel Product Configurator server deployment mode. Except where noted, set these parameters on the Application Object Manager component.

Parameter Name	Display Name	Data Type	Default Value	Description
eProdCfgRemote	Product Configurator - Use Remote Service	Boolean	False	Setting to determine whether Siebel Product Configurator is running on a different server from the Application Object Manager. On the Application Object Manager: set it to True when running a dedicated Siebel Product Configurator server. On the dedicated Siebel Product Configurator server: leave this set to False.
eProdCfgServer	Product Configurator - Remote Server Name	Text	None	Name of the Siebel Server on which you are running a dedicated Siebel Product Configurator server. If you are using multiple dedicated Siebel Product Configurator server, then separate the entries with semicolons (;).
eProdCfgTimeOut	Product Configurator - Time Out of Connection	Integer	20	Setting in seconds that determines for how long the Siebel Server would try to initiate a connection with the remote Siebel Product Configurator server before returning an error to the user.
eProdCfgKeepAliveTime	Product Configurator - Keep Alive Time of Idle Session	Integer	900	Setting in seconds to determine the maximum interval of inactivity during a configuration session. If the interval of inactivity reaches this value, then the user session is ended and the worker returns to the pool. If this parameter is not set, then an infinite interval is assumed. Set this parameter on the Application Object Manager only. It does not apply on the remote Siebel Product Configurator server.

Parameter Name	Display Name	Data Type	Default Value	Description
				Note: On the remote Siebel Product Configurator server (eProdCfgObjMgr component), set the parameter ConnIdleTime to a value like eProdCfgKeepAliveTime plus 1 second.

Siebel Workflow Deployment Planning

Siebel Workflow lets you define, manage, and enforce your business processes or workflows. It allows you to design complex workflow processes and automate the enforcement of business policies and procedures. This module includes the following:

- **Workflow Processes.** This module lets you define your company's business processes using a familiar flowcharting interface. A workflow process consists of one or more process steps, such as start steps, subprocesses, decision points, and tasks. You configure workflow processes using the Process Designer in Siebel Tools.
- **Workflow Policies.** This module lets you define policies that can act as triggers to execute a process. A policy consists of conditions and actions. When policy conditions are met, the policy action executes the relevant process.
- **State Models.** This module is used for defining business object states and state transitions.

For information about using and administering Siebel Workflow, see *Siebel Business Process Framework: Workflow Guide* .

Each user request to the Workflow Process Manager starts a new thread. However, sessions for Object Manager components (such as EAI Object Manager or Application Object Manager) that might invoke workflow processes are cached and reused for subsequent requests. When you size a deployment of Siebel CRM applications, the maximum number of workflow tasks that you expect to have active at a given time helps determine the maximum number of Object Manager sessions created for Siebel applications.

The exact CPU and memory consumption of each task depends on the actions performed in your workflow processes. To estimate CPU and memory consumption in your production environment, run a single task, measure its resource consumption, and make an estimation based on your maximum concurrent sessions. Take session caching into account when making these measurements.

If you need a large number of sessions, then you might want to run Workflow Process Manager on multiple Siebel Server computers. You can then load-balance requests across the Siebel Servers. If you plan to run a significant number of tasks per server (such as 100 or more), then you might also want to run multiple multithreaded processes.

If you are going to run several different types of workflows, then run each type in a separate process. Doing so makes it easier to monitor the overall CPU and memory usage of each process type.

The number of multithreaded processes and the number of tasks for each process are controlled through the parameters MaxMTServers (Maximum MT Servers), MinMTServers (Minimum MT Servers), and MaxTasks (Maximum Tasks).

These parameters are for each Siebel Server. For example, MaxMTServers refers to how many multithreaded processes to run on each Siebel Server computer.

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

Planning Batch Processing When Using Siebel Remote

Long-running batch jobs can create transaction gaps in the Master Transaction Log for the Siebel Remote module for Oracle's Siebel CRM. If the wait-time for the missing transactions expires, then the Transaction Processor component skips the missing transactions. The skipped transactions are not routed to mobile users.

Batch jobs could be performed by Siebel Assignment Manager, Siebel EIM, or other components.

For more information to help you to understand gaps in the Master Transaction Log, see 477585.1 (Article ID) on My Oracle Support. This document was previously published as Siebel Technical Note 499.

For an example with Siebel Assignment Manager, gaps in the Master Transaction Log can occur as follows:

1. Assignment Manager is processing a batch of transactions.
2. Assignment Manager obtains a group of transaction IDs. These are issued in numeric, sequential order.
3. Assignment Manager then commits these transactions. This process takes several minutes.
4. In the meantime, another process obtains a transaction ID.
5. The process commits the transaction and writes it to Siebel Remote's Master Transaction Log. A sequence gap is created because the Assignment Manager transactions have not yet been written to the Master Transaction Log.
6. Transaction Processor detects the gap and waits a specified period called the wait-time (default is 600 seconds).
7. The wait-time expires before Assignment Manager completes the commit operation and writes the missing transactions to the Master Transaction Log.
8. When the wait-time expires, Transaction Processor skips the missing transactions and moves on to the transaction from the other process.
9. Transaction Processor logs information about the missing transactions. The Assignment Manager transactions are not routed to mobile users, even though they are later written to the Master Transaction log.

This topic contains the following information:

- *Conditions That Can Cause Missed Transactions*
- *Avoiding Missed Transactions*

Conditions That Can Cause Missed Transactions

The following conditions in Assignment Manager can cause increased commit times. Longer commit times increase the risk that the Transaction Processor wait-time expires before the commit occurs and that Transaction Processor fails to process all of the transactions in the transaction log.

This topic is part of *Planning Batch Processing When Using Siebel Remote*.

Increasing the Assignment Manager Batch Commit Parameter

The default batch commit size (BatchSize) for Assignment Manager is 100. After processing 100 rows, transactions are committed to the database. If the batch commit size is increased, then this increases the risk of exceeding the wait-time.

Increased Number of Batch Assignment Threads

When multiple Assignment Manager threads are logging transactions, this creates a latency in accessing the transaction log table. This latency increase the risk of exceeding the wait-time.

Complicated Assignment Rules

When Assignment Manager has to resolve complicated assignment rules, this can increase commit times. Longer commit time together with the preceding conditions can increase the risk of exceeding the wait-time.

Avoiding Missed Transactions

To avoid exceeding the Transaction Processor wait-time during batch processing, adopt the following recommendations. Experiment with applying them in combination to achieve the best performance while minimizing the risk of exceeding the wait-time.

This topic is part of *Planning Batch Processing When Using Siebel Remote*.

Monitor the Transaction Processor Logs

As you apply the recommendations described as follows, use the Transaction Processor logs to see the result and help you to optimize system performance.

Transaction Processor writes these warning messages to its log file when it skips transactions:

```
GenericLog: GenericError: 0003-11-18 17:04:51
WARNING: A transaction gap has been detected after transaction 122.
Probable Cause: There maybe long-running transactions in your system which are not
committing transactions within the specified duration (600 sec)
Recommendation: Reduce the batch size of your transactions. This will allow the
transactions to be committed to the database within the wait-time window.
```

If skipped transactions occur while a batch job is running, then investigate the cause. The skipped transactions might not have been routed to mobile users. If so, then the mobile users might have to re-extract the database.

Set a Lower Batch Size for Assignment Manager

Setting a lower BatchSize value reduces the number of records processed before each commit. Reducing this figure reduces the commit times and the risk of exceeding the wait-time. If your performance goals require you to increase the BatchSize parameter, then do so only after analyzing the number of Assignment Manager threads that you have under average and peak workloads. The fewer Assignment Manager threads, the higher that you can set the BatchSize parameter.

You can get performance statistics on threads by raising the Assignment Manager log level. For information about raising the log level, see *Siebel System Monitoring and Diagnostics Guide* .

Serialize Batch Jobs

Consider staggering the start time of batch jobs. Running batch jobs in staggered or serial order can reduce the risk of exceeding the wait time.