Deployment Planning Guide

Version 7.7, Rev. A

May 2004

Siebel Systems, Inc., 2207 Bridgepointe Parkway, San Mateo, CA 94404

PRODUCT MODULES AND OPTIONS. This guide contains descriptions of modules that are optional and for which you may not have purchased a license. Siebel's Sample Database also includes data related to these optional modules. As a result, your software implementation may differ from descriptions in this guide. To find out more about the modules your organization has purchased, see your corporate purchasing agent or your Siebel sales representative.

Proprietary Information

Siebel Systems, Inc. considers information included in this documentation and in Siebel eBusiness Applications Online Help to be Confidential Information. Your access to and use of this Confidential Information are subject to the terms and conditions of: (1) the applicable Siebel Systems software license agreement, which has been executed and with which you agree to comply; and (2) the proprietary and restricted rights notices included in this documentation.

Contents

Chapter 1: What’s New in This Release

Chapter 2: Siebel Architecture Overview

Chapter 3: Siebel Infrastructure Planning

Chapter 4: High-Availability Deployment Planning

Chapter 5: Load Balancing and Resilient Processing Planning

Chapter 6: Managing Siebel Load Balancing

Chapter 7: Managing Third-Party Load Balancing

Chapter 8: Server Clustering Planning

Chapter 9: Data Integrity and Capacity Planning

Chapter 10: Application-Level Deployment Planning

Chapter 11: Siebel Client Deployment Planning

1 What’s New in This Release

What’s New in Deployment Planning Guide, Version 7.7, Rev. A

Table 1 lists changes in this version of the documentation to support release 7.7 of the software.

Table 1. What’s New in Deployment Planning Guide, Version 7.7, Rev. A

Topic	Description
“Mapping Siebel Deployment Elements to Platforms” on page 27	Added guidelines for topology planning and new table that describes deployment schemes.
“About Setting Siebel Server Task Loads” on page 55	New topic.
“Troubleshooting Siebel Load Balancing” on page 65	New topic.
“Verifying IP Access to Siebel Servers” on page 68	New topic.
“Verifying Load Balancing Port Access for Siebel Servers” on page 68	New topic.
“Manually Rebalancing Siebel Server Loads” on page 69	New topic.
“Setting the Load Balancer Connection Time Out” on page 74	New topic.
“Monitoring Servers with Siebel Load Balancing”	Topic deleted. SWSE Statistics page is not supported in release 7.7.

What’s New in Deployment Planning Guide, Version 7.7

Table 2 lists changes in this version of the documentation to support release 7.7 of the software..

Table 2. New Product Features in Deployment Planning Guide, Version 7.7

Topic	Description
“About Load Balancing” on page 45	■ A load balancing module has been added to the Siebel Web Server Extension (SWSE). ■ In addition, Siebel has certified several third-party HTTP load balancers. ■ Resonate Central Dispatch is no longer supported.
“About the Siebel Connection Broker” on page 52	A new server component, Siebel Connection Broker, distributes new session requests to multiple instances of Application Object Managers running on the same Siebel Server.

2 Siebel Architecture Overview

This chapter includes the following topics:

Building Blocks of a Siebel Deployment

Figure 1 on page 10 shows an example of the elements in a Siebel deployment. A brief description of these elements appears in Table 3 on page 11.

Specific database and operating system platforms, as well as certain combinations of them, may not be supported by the current release. For a list of all operating system platforms and RDBMS products supported by this release, consult *System Requirements and Supported Platforms* on Siebel SupportWeb*.*

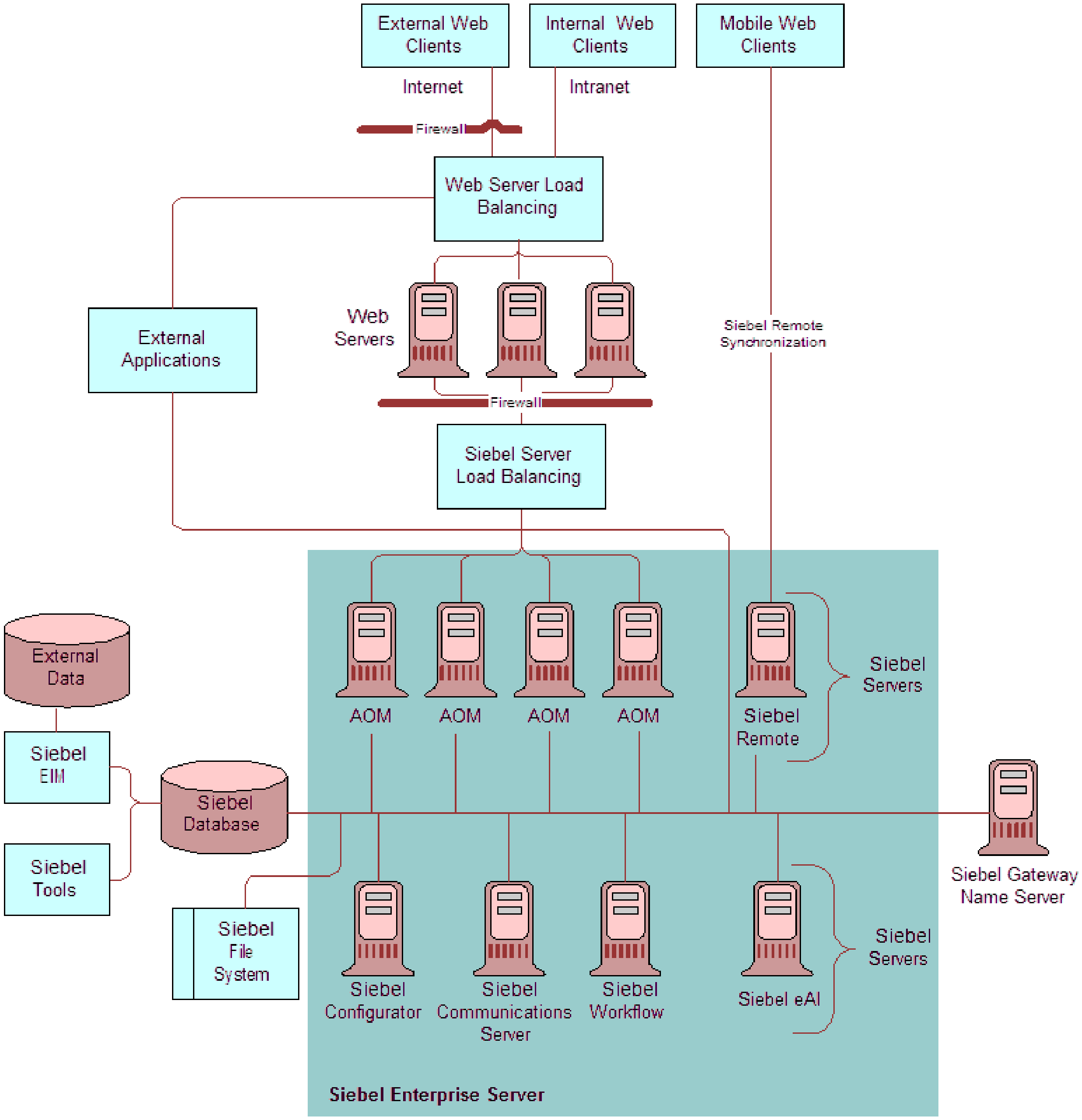

Figure 1. Example of a Siebel Deployment

Table 3. Siebel Deployment Elements

Entity	Description
Siebel Web Clients	Includes the following client types: ■ Siebel Web Client ■ Siebel Mobile Web Client ■ Siebel Wireless Client ■ Siebel Handheld Client
Siebel Web Server Extension (SWSE)	Installed on third-party Web server. Identifies requests for Siebel data and forwards them to the Siebel Servers. Receives data from Siebel Servers and helps format it into Web pages for Siebel clients.
Siebel Load Balancing	The two options for Siebel Server load balancing are Siebel load balancing and third-party HTTP load balancers. Siebel load balancing is part of the Siebel Web Server Extension (SWSE). When you install the SWSE, the installation wizard prompts you for information about configuring load balancing. Figure 1 on page 10 shows a third-party HTTP load balancer.
Siebel Enterprise Server	A logical grouping of Siebel Servers that connect to one database. Allows management of Siebel Servers as a group.
Siebel Servers	Application servers that provide both user services and batch mode services to Siebel clients.
Siebel Gateway Name Server	Functions as a name server and stores Siebel Server configuration information.
Siebel Database	Stores database records. Includes third-party RDBMS software and Siebel tables, indexes, and seed data.
Siebel File System	Shared file system directory that stores the data and physical files used by Siebel clients and Siebel Enterprise Server.
Siebel deployment	All of the elements required to deploy Siebel applications. This includes the Siebel Enterprise Server, Siebel Servers, Siebel Database, Siebel Gateway Name Server, Siebel Web Server Extension, and related components such as third-party HTTP load balancers.
Siebel Enterprise Integration Management (EIM) and Siebel Enterprise Application Integration (EAI)	Allows import and export of data from other databases to Siebel Database.
Siebel Tools	Provides an object-oriented, Windows-based environment for developing or modifying Siebel applications, business services, and other Siebel objects.

About Siebel Web Clients and Web Server Extension

There are several types of Siebel Web Clients:

Siebel Web Client

- **Installed software**. No additional application software required on the client. Requires only a Web browser.
- **Application connection**. Through a Web server to the Siebel Enterprise Server. Applications run on Siebel Server and forward pages to the client.
- **Database connection**. Through the Siebel Server to the remote Siebel Database. No Siebel Database or database client installed on the client.

Siebel Web Client runs in a standard browser on the end user's client computer. ActiveX controls and JavaScript routines are downloaded to the browser automatically when it runs Siebel applications in Siebel high interactivity mode. The browser connects through a Web server to the Siebel Server, which executes business logic and accesses data from the Siebel Database. Only the user interface layer of the Siebel eBusiness Applications architecture resides on the user computer.

Wireless Client. Siebel Wireless is a modified Siebel Web Client that runs on a mobile device. Users can view, edit, and create records in the Siebel Database through a wireless connection between a mobile device and a Web server. An internet-enabled mobile phone, personal digital assistant or other device communicates using wireless application protocol (WAP) to a wireless gateway server. The wireless gateway server translates HTTP messages to WAP. The Siebel interface is rendered on the mobile device using wireless mark-up language (WML). Specific XML- and HTTP-based wireless browsers are also supported. For a list of Siebel eBusiness Applications that support wireless access, see the *Siebel Wireless Administration Guide*. For a list of supported wireless browsers, see *System Requirements and Supported Platforms* on Siebel SupportWeb*.*

Siebel Mobile Web Client

- **Installed software**. Windows-based software containing Siebel applications and related services is installed on each client.
- **Application connection**. Applications run on each client. Applications display in a Web browser.

- **Database connection**. A Siebel Database and Siebel File System are installed on each client. Applications access the client’s local database.

  Users periodically synchronize the client’s local database and Siebel File System with a remote Siebel Enterprise Server’s Siebel Database and Siebel File System. Users synchronize data using Siebel Remote. Siebel Remote allows users to connect directly to the remote Siebel Database, and Siebel File System without going through the Enterprise Web servers or Siebel Servers.

  The Mobile Web Client runs in a standard browser on the end user’s client computer, such as a laptop.

Siebel Handheld Client. The Siebel Handheld Client is a streamlined version of the Siebel Mobile Web Client. It includes only the functionality required by end users’ field technicians. The Siebel Handheld Client supports the same data relationships, the same configuration in Siebel Tools, and much of the same functionality as the Siebel Mobile Web Client. Siebel Handheld runs on devices that support the Windows CE operating system.

Siebel Web Server Extension (SWSE)

The Siebel Web Server Extension (SWSE) is a plug-in for third-party Web servers. It identifies requests for Siebel information 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 SWSE helps complete the composition of the Web page for forwarding to the client.

Included in the SWSE is the Siebel load balancing module. This module provides round-robin load balancing for Application Object Managers running on Siebel Servers.

All language packs installed on Siebel Servers must also be installed on your Web servers. However, you do not need to install all language packs on all Web servers. If you do not install all languages on all Web servers, you must provide a means for routing user requests to Web servers that have the correct language support.

About Siebel Enterprise Server and Siebel Server

The Siebel Enterprise Server is a logical grouping of Siebel Servers that connect to one Siebel Database. The Siebel Servers in a Siebel Enterprise Server are configured, managed, and monitored as a single logical group, allowing the Siebel administrator to start, stop, monitor, or set server parameters for all Siebel Servers within the Siebel Enterprise Server.

Siebel Server

The Siebel Enterprise Server is composed of one or more Siebel Servers. Siebel Servers function as application servers and are composed of server components. Each server component performs a defined function.

Server components or groups of components determine what applications and services a Siebel Server supports. Components run in one of several modes:

- **Interactive mode.** Interactive components start tasks automatically in response to user requests. The tasks end when the user ends the session. Examples of interactive mode tasks are the Synchronization Manager and all Application Object Managers (AOMs).
- **Background mode**. Background components handle background processing tasks. Typically, background tasks are called by interactive mode tasks. Background tasks run until explicitly shut down. Examples of background tasks are Transaction Router and Workflow Monitor Agent.
- **Batch mode**. Batch mode components handle processing of asynchronous work requests. When the task is complete, the component exits. Examples of batch mode components are Database Extract and Enterprise Integration Manager (EIM).

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

Other Siebel Server components provide additional functionality besides application support. This includes the following:

- Siebel Mobile Web Client synchronization.
- Integration with legacy or third-party data.
- Automatic assignment of new accounts, opportunities, service requests, and other records.
- Workflow management.
- Document generation.
- Siebel Connection Broker (SCBroker). This server component provides load balancing for multiple Application Object Managers (AOMs) running on the same Siebel Server.

Implementation. The Siebel Server runs as a system service under Windows and a process under UNIX. This system service or process monitors and controls the state of all 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 mode and batch mode components can be configured to run as multiple processes or as multithreaded processes. Background mode components run as multiple processes only.

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

Application Object Manager (AOM)

One of the most important types of server components is the Application Object Manager (AOM). These server components run in interactive mode. They process user requests and are application- or service-specific. For example, the Siebel Employee Relationship Management component group contains the Employee Relationship Object Manager. This Application Object Manager provides the session environment in which this application runs.

AOMs also contain a data manager, and the Siebel Web Engine. When an AOM receives a user request to start an application, it does the following:

- 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 and SQL query and forwards it the database server.

- 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 UI for the data. The Siebel Web Engine then forwards the Web pages to the Siebel Web Server Extension on the Web server.

Implementation. An Application Object Manager (AOM) server component is implemented as a multithreaded process on the Siebel Server. At runtime, a parent process starts one or more AOMs as multithreaded processes, according to the AOM configuration. The terms *multithreaded server* or *MT server* are alternative terms for the multithreaded process.

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

Each AOM task uses the Siebel Server to communicate with the Siebel Database, the Web server (through the SWSE), and other Siebel Enterprise Server components:

- Communication with the Siebel Database uses ODBC database connections. Database connections can be managed and tuned for optimal performance. You can optionally configure connection sharing for database connections.
- Communication with the Siebel Web Server Extension uses SISNAPI (Siebel Internet Session API), a Siebel messaging format that runs on top of the TCP/IP protocol. SISNAPI connections can be configured to use encryption and authentication based on Secure Sockets Layer (SSL).
- 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 Web server. SCBroker forwards these requests to AOMs.

About Siebel Gateway Name Server

The Siebel Gateway Name Server serves as the dynamic address registry for Siebel Servers and components. At start up, a Siebel Server within the Siebel Enterprise Server stores its network address in the Gateway Name Server's nonpersistent address registry.

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

The Gateway Name Server also includes a persistent file (siebns.dat) containing Siebel Server configuration information, including:

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

As this information changes, such as during the installation or configuration of a Siebel Server, it is written to the configuration file on the Name Server.

In a production environment, there can be only one Name Server installed per machine. Do not share the same Name Server for your development, test, and production environments.

Language Pack Installation. You do not need to install all the languages that your Siebel Deployment may run on the Siebel Gateway Name Server. However, the Siebel Gateway Name Server installation includes utilities used for Siebel Server administration. Siebel administrators only see some server administration error messages in the languages that have been installed on the Siebel Gateway Name Server.

About Siebel File System

The Siebel File System is a shared file system directory. The Siebel File System stores document files, Siebel Configurator models, Web template definitions, and other files not appropriate for database storage.

The File System Manager, a Siebel Server component, manages all file requests from other Siebel Server components, such as Application Object Managers.

For more information on the Siebel File System, see the *Siebel System Administration Guide*.

About Siebel eBusiness Applications Integration (EAI)

Siebel EAI provides components for integrating Siebel eBusiness Applications 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. It also includes tools for cross application integration through Universal Applications Network (UAN).

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, BizTalk, and OLE DB; Sun Microsystems Java and J2EE; XML, and HTTP, and many other standards.

These 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 on EAI, see *Overview: Siebel eBusiness Application Integration Volume I*.

About Siebel Enterprise Integration Manager (EIM)

Siebel Enterprise Integration Manager (EIM) manages the bidirectional exchange of data between the Siebel Databases and other corporate databases. This exchange is accomplished through intermediary tables called EIM tables (in earlier releases, these tables were known as Interface Tables). The EIM tables act as a staging area between the Siebel application database and other databases.

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

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

About Siebel Tools

Siebel Tools is a Windows-based, object-oriented development environment for creating and customizing Siebel eBusiness Applications. Siebel Tools also provides a means for integrating 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 with different purposes:

- Physical User Interface (UI) Layer: Templates and tags that render the UI.
- Logical User Interface Objects Layer: Presentation of data (UI).
- 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. This means, for example, that you can make changes to a Siebel application without changing the underlying database structure. Similarly, you can extend the Siebel Database schema without affecting the Siebel application.

For additional information on Siebel Tools see the *Configuring Siebel eBusiness Applications*.

Example of User Request Flow in a Siebel Deployment

Figure 2 illustrates how a user request is processed within the Siebel eBusiness Applications architecture.

In the diagram, there are two types of load balancing:

- **Web server load balancing**. Web client requests are forwarded through a load balancer to multiple Web Servers.
- **Siebel Server load balancing**. Web servers forward user requests to a third-party HTTP load balancer for distribution to Siebel Servers.

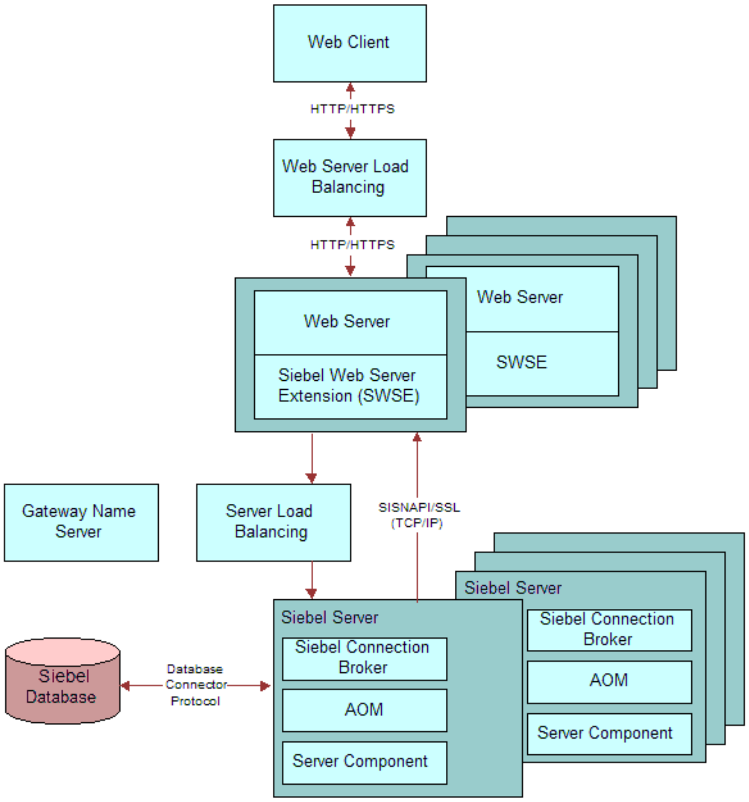

Figure 2. Generic User Request Flow in Siebel eBusiness Applications

A typical Siebel client request flows from the user’s Siebel Web Client through the system, and back again, following the general flow outlined below.

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 may go through a network router, proxy server, cache engine, or other mechanism.

3 If present, Web server load balancing software evaluates the request and determines the best Web server to forward the request to. It then forwards the request to a Web server.

4 The Web server receives the HTTP request, determines that it is a Siebel application request, and forwards the request to the Siebel Web Server Extension (SWSE) installed on the Web server.

5 The Siebel Web Server Extension parses the HTTP message and generates a SISNAPI message, based on the content of the HTTP message. SWSE also parses the incoming cookie or URL to obtain the user session ID.

   - If using Siebel load balancing, SWSE forwards the request to a Siebel Server in round-robin fashion.

   - If using a third-party HTTP load balancer, SWSE forwards the request to the load balancer. The load balancer uses user-configured routing rules to forward the request to a Siebel Server.

   SISNAPI (Siebel Internet Session application programming interface), is a messaging format that runs on top of the TCP/IP protocol. It is used for network communication between Application Object Managers (AOMs) and SWSE.

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

   The database request goes through the database connection, using a protocol format that is specific to the database connector.

7 The database executes the SQL statement and returns data back to the AOM. The AOM forwards the message to the Web server that originated it. If using a third-party HTTP load balancer, the message may go through the load balancer before reaching the Web server.

8 The SWSE on the Web server receives the SISNAPI message, and translates it back to HTTP. It then forwards the HTTP message to the Web server. The message is now in the form of Web page content.

9 The Web server load balancer, if present, then 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.

3 Siebel Infrastructure Planning

This chapter explains how to plan the infrastructure of your Siebel deployment.

The chapter contains the following topics:

Process of Infrastructure Planning

This process shows you how to determine the Siebel infrastructure requirements for a production environment. Along with a production environment, you should also plan for a software development and a test environment.

Use the following steps to plan your Siebel deployment infrastructure:

1. "Determining How the System Will Be Used" on page 22
2. "Defining Data Flows and Integration Requirements" on page 23
3. "Determining Database Requirements" on page 24
4. "Mapping Business Requirements to Siebel Server Components" on page 25
5. "Defining High-Availability Policies" on page 26
6. "Mapping Siebel Deployment Elements to Platforms" on page 27
7. "Determining Network Requirements" on page 30
8. "Defining a Test and Transition Plan for the Siebel Deployment" on page 31

Determining How the System Will Be Used

This infrastructure planning step identifies what tasks users will perform when using the system.

Examples are completing a customer order, adding a contact, and creating a quote. Later in the planning process, you will map these tasks to specific Siebel applications and functions.

This topic is a step in "Process of Infrastructure Planning" on page 21.

To determine how the system will be used

1. Define user types.

   For each business location, identify user types. Organize this list by the functional areas that participate in key business processes.

   For example, 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.

   Include application developers and integrators, system administrators, and application administrators in your list of user types.

2. Identify tasks by user type.

   For each user type, identify all the tasks the user type will perform using the system. Start with each key business process and map its steps to tasks. This allows you to verify that business processes are being correctly automated.

3. Identify background tasks.

   If your business operation includes background tasks, list these as well. Background tasks are those that the system performs, 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 will complete an average of 12 customer orders per day and a maximum of 20 per day. Table 4 shows an example of how you would list transaction volumes.

Table 4. Denver Transaction Volumes

User Type	Number	Task	Avg. Vol./ Day	Max. Vol./ Day
Call Center Agents	25	1. Inbound customer order	300	500

Defining Data Flows and Integration Requirements

This infrastructure planning step identifies how data will flow to and from the Siebel 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 step identifies where the master copy of data records will reside. It also identifies the data interchange requirements for applications.

This topic is a step in "Process of Infrastructure Planning" on page 21.

To identify data flows and transaction volumes

1. Identify business data.

   List the types of business data that will flow through the system. 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 data requirements of legacy applications.

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

4. Identify data formats and transformations.

   For each legacy application that will send or receive data from the Siebel Application, identify the required data formats. Specify in detail all data transformation requirements.

5. Map the data flows.

   Create a model that shows all major business data flows. The model should include all data sources, repositories, and key business applications.

Figure 3 shows an example of a model of a data flow. The example shows a call center running Siebel eCommunications. The company maintains an ERP database and a phone number database separately from the Siebel database, which contains customer information.

Siebel eCommunications sends XML messages containing customer orders to the order fulfillment application, and receives order fulfillment status through an inbound HTTP adaptor. Siebel eCommunications also queries the Phone Number Management System for available phone numbers in real-time. The Phone Number Database then receives assigned phone numbers from the Siebel Database using Siebel EIM.

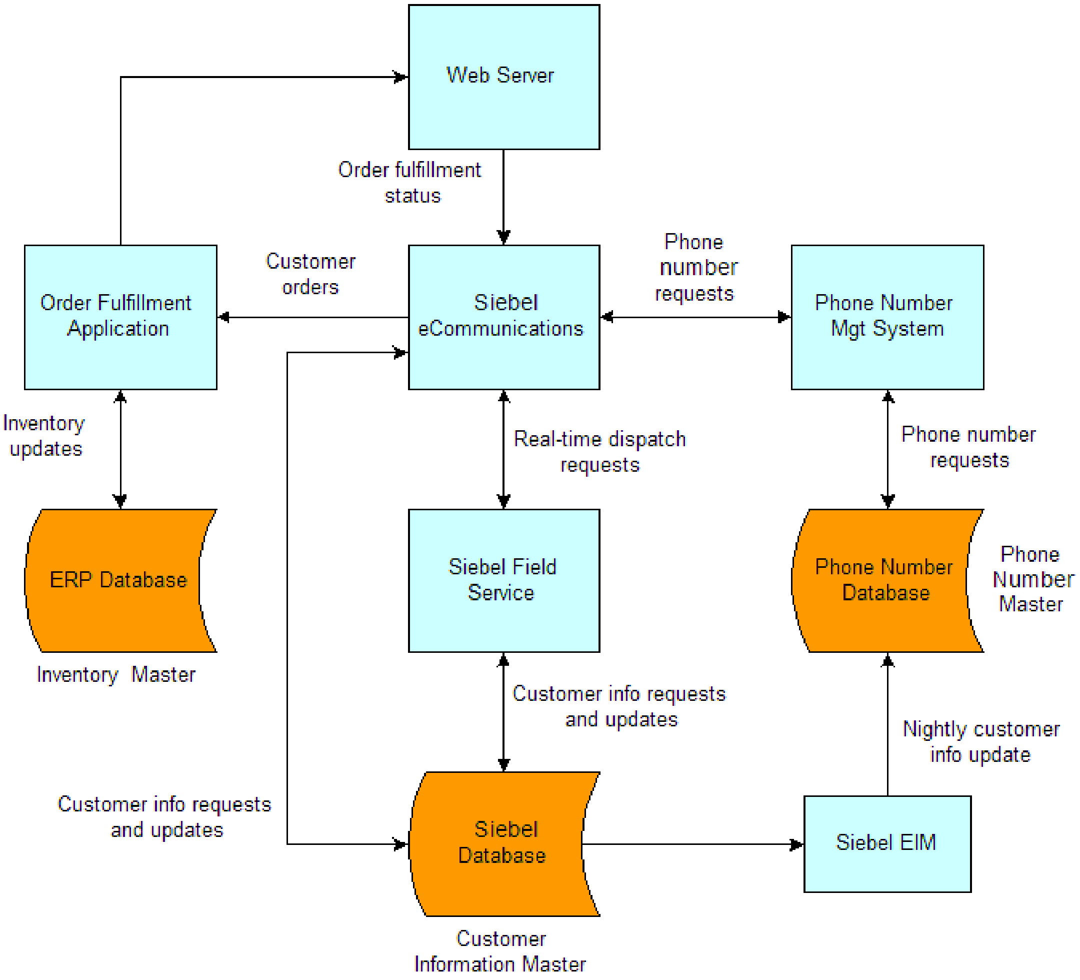

Figure 3. Example of a Data Flow Model

Determining Database Requirements

This infrastructure planning step identifies database requirements for the Siebel deployment.

You should have already identified the types of data that will be stored in the Siebel database. This step maps that data to key database characteristics. This allows you to estimate database size requirements and expected growth.

This topic is a step in "Process of Infrastructure Planning" on page 21.

Begin by defining general requirements:

- What are the types of records that will be stored? What specific fields will each record contain?
- What is the volume of each record? How many records of each type will be processed each hour? Each day? Each year? Group this information by business location.
- Determine how record volumes map to specific Siebel tables. Contact Siebel Expert Services for information on mapping records to Siebel tables.
- How much space will database indices occupy? Typically, indices require as much space as the database. For example, a 50GB database will require an additional 50GB of indices.
- 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 will be assigned to each customer account.
- Number of employees that will be assigned to each account.
- Number of contacts that will be assigned to each account.
- Number of attachments that will be assigned to a record.
- Number of activities that will be associated with each account.
- Will opportunities, quotes, or orders be stored?
- Will product data be stored?
- Will Siebel Remote be used?

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

Mapping Business Requirements to Siebel Server Components

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

This topic is a step in "Process of Infrastructure Planning" on page 21.

Begin by listing the Siebel applications that users will run. For each application identify the associated Application Object Manager (AOM). If you are deploying internationally, list the language-specific AOMs you will need.

Many AOMs 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 AOMs and of the Siebel Server.

A common problem when Siebel Expert Services does an implementation readiness review at customer sites, is that second tier component requirements are not correctly identified. To avoid this, work closely with the application development team to identify these components.

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

Defining High-Availability Policies

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

This topic is a step in "Process of Infrastructure Planning" on page 21.

Siebel Servers

For each business location, assess the impact of losing each server component. Think in terms of the component failing rather than the hosting platform. A common problem when Siebel Expert Services does implementation readiness reviews at customer sites, is that failure of individual server components has not been adequately analyzed. The result is that server components important to normal application function are not recognized as single points of failure.

After you complete this analysis, define high-availability policies for all 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 will run 24X7, and the maximum acceptable downtime is 30 minutes. The company also decides the maximum time it can accept degraded performance is one hour.

Finally, at each business location, list all 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 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 RAID arrays 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 best-practice guidelines 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 4 rollback segments for each 20 to 40 users. The size of rollback extents should be 100K/100K. If you are using Siebel EIM, create several additional, large rollback segments to support EIM loads.

Siebel Gateway Name Server

The Siebel Gateway Name Server maintains the configuration information for all Siebel Servers in all the Siebel Enterprise Servers it manages. Loss of the Siebel Gateway Name Server due to a disk failure could bring your Siebel deployment to a halt while the system is restored.

It is strongly recommended that you install a redundant disk array (RAID) or some other type of redundant disk configuration on your Siebel Gateway Name Server.

Mobile Users

A Siebel Server temporarily stores transaction files that move to and from Siebel Remote mobile users. The loss of these files will result in the need to re-extract the database for all affected mobile users. (Siebel Remote supports synchronization of data between Siebel Mobile Web Clients and the Siebel Database Server through a dial-up connection.)

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

Mapping Siebel Deployment Elements to Platforms

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

Criteria. Mapping Siebel 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

Prerequisites. Review the following information, developed in previous steps:

- Database requirements. See “Determining Database Requirements” on page 24
- Required Siebel Server components. See “Mapping Business Requirements to Siebel Server Components” on page 25
- High-availability policies. See “Defining High-Availability Policies” on page 26

This topic is a step in “Process of Infrastructure Planning” on page 21.

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 (AOM) on each Siebel Server. This is a common practice, but not a requirement. Customers should contact Siebel Expert Services for information on how to size Siebel Servers.

2. Identify which Siebel Server components can be collocated. Distribute these across platforms in a way that evenly distributes workload.

   Object managers generate a fairly level workload that does not include large spikes. In contrast, large workflow processes generate uneven workloads that include large spikes when executing workflow steps.

   When deciding which server components to collocate, observe the following best practices:

   - Object managers should be collocated with other object managers.
   - Object managers should not be collocated with workflow processes. This will minimize the impact of workflow processes on user application performance.

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).

4. Identify additional hardware required to comply with security policies. For example, do you need to install additional firewalls or a proxy server? Do you need to install LDAP servers?

5. Use average and peak workload information to determine how many Web servers are needed.

6. Create a diagram of the Siebel deployment that shows all the platforms and the distribution of Siebel Servers. Use the diagram to do the following:

   a. Verify that all 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 will cause unacceptable impacts.

   For example, you have one Web server. All your inbound customer orders must go through it and then to one HTTP inbound adapter. If the Web server or the inbound adapter fails, 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 (AOMs) run on one group of machines, workflows on a second group of machines, and remote user synchronization on a third group. Give the AOM servers names starting with APP, the workflow servers names starting with WF, and the Siebel Remote servers names starting with REM.

   Each group will display together in Server Manager. This simplifies server administration.

Topology Planning Guidelines

Use the following guidelines for topology planning:

- A single Siebel Gateway Name Server can be used to manage multiple Siebel Enterprise Servers.
- A Siebel Enterprise Server can belong to one and only one Siebel Gateway Name Server.
- 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. This includes multiple Application Object Manager types, each with their own SRF.
- Table 5 lists supported deployment schemes.

Table 5. Deployment Schemes

Deployment Scheme	Recommended?
Single Siebel Gateway Name Server and multiple Siebel Enterprise Servers running on a single machine or UNIX hardware partition. Each Siebel Enterprise Server has its tableowner and database server.	No. If the Siebel Gateway Name Server fails, it would adversely affect all the Siebel Enterprise Servers. Failure of one UNIX partition could adversely affect all the Siebel Enterprise Servers. If one Enterprise Server requires an upgrade, all Enterprise Servers are affected.
Running multiple Siebel Gateway Name Servers on a single UNIX hardware partition or on a single unpartitioned machine.	No. Very difficult to set up. Requires manual workarounds. Requires manually editing registry on Windows platforms.
Multiple Siebel Enterprise Servers sharing a DBMS tableowner.	No. Each Siebel Enterprise Server is required to have its own set of tables.
Multiple Siebel Enterprise Servers, each with their own DBMS tableowner and sharing the same instance of the DBMS executable.	No. If the one Database Server goes down, all of the Siebel Enterprise Servers will go down. Too much dependency on one machine.
Siebel Enterprise Server hosting multiple Siebel Servers within a single hardware partition.	No. No additional scalability, throughput, or performance benefits to this configuration. Managing multiple Siebel Servers within a single hardware partition also requires more Siebel administrator time.
Siebel Enterprise Server hosting multiple Siebel Servers, with each Siebel Server on its own machine or UNIX hardware partition (multiple partitions on each UNIX server machine).	Yes. This is the most common way to deploy Siebel version 6.x and later.

Table 5. Deployment Schemes

Deployment Scheme	Recommended?
A single Siebel Server managing multiple applications. Each Application Object Manager type having its own copy of the SRF.	Yes. This is a common deployment scheme. It allows you to segment functionality per process. If multiple SRFs are used for the Object Managers in a Siebel Enterprise Server, the SRF used for common components such as Workflow Process Manager, EAI Object Managers, and so on should have the common objects needed for proper processing. It is critical to have all SRFs in sync. The Siebel Repository in the production DBMS must also be in sync with the SRFs.
Installing the Siebel Gateway Name Server, each Siebel Server, and the database all on different operating systems.	Yes. This is supported for Siebel 7.x. However, you should try 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 you need only runs on another. For information on supported operating systems for Siebel deployments, see *System Requirements and Supported Platforms* on Siebel SupportWeb.

Determining Network Requirements

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

This topic is a step in "Process of Infrastructure Planning" on page 21.

To determine network requirements

1. Use the information on 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 deployment.
2. Determine whether you will use data encryption. If so, define data encryption policies. Then add data encryption protocols to the diagram created in the previous topic ("Mapping Siebel Deployment Elements to Platforms" on page 27).
3. Define firewall requirements. If you will create a network DMZ, also define requirements for proxy servers and other items that will be installed in the DMZ.

   Include network address translation (NAT) and HTTPS requirements.

4. Analyze interactions and dependencies between networking components.

   For example, you plan to use HTTP/SSL between browsers and Web servers. You also plan to install a Web server load balancer. Typically, Web server load balancers cannot do HTTP, URL-based load balancing unless the load balancer has an integrated SSL accelerator.

   An SSL accelerator allows SSL connections to be terminated at the Web server load balancer. This allows the load balancer to parse packet information and perform HTTP, URL-based load balancing.

5. Write down the virtual IP (VIP) address used by Web server and Siebel Server load balancers. Make sure the requesting component is set up to access the VIP. Furthermore, make sure that firewalls are configured to allow VIP traffic to go through.

6. Select port numbers for all network components that listens on TCP ports.

   This includes Web server load balancers, Web servers, Siebel Server load balancers, Siebel Servers, and server clusters. For a full description of Siebel Server components that require a port number assignment, see *Siebel System Administration Guide*.

   The default TCP port number for Web server-to-Siebel Server traffic is 2321. This is the port number of the Siebel Connection Broker (SCBroker). The port number is configurable.

   If a third-party HTTP load balancer is deployed, then it must be set up to communicate to Siebel Servers using the SCBroker port.

   The Siebel Gateway Name Server and Siebel Servers cannot be assigned port numbers higher than 32767.

   If Siebel load balancing is deployed, then the load balancing configuration file must reference this port number. See Chapter 5, "Load Balancing and Resilient Processing Planning."

   Also, verify that firewalls are configured to communicate with the correct TCP ports.

7. Consider any other factors that may affect networking connectivity.

Defining a Test and Transition Plan for the Siebel Deployment

Defining a test plan to verify that the proposed deployment infrastructure functions correctly and is sized correctly is critically important. Equally important is defining a plan that transitions the Siebel deployment to production.

This topic is a step in "Process of Infrastructure Planning" on page 21.

Observe the following best-practices guidelines for testing the Siebel deployment and transitioning it to production.

- **Separate production environment**. Keep the development and test environments physically separate from the production environment. Development and test activities should never be conducted on the production Siebel Database and preferably not on the production database server.

- **Server stress testing**. Test Siebel Enterprise Server performance under average and peak workloads. Siebel Expert Services finds that performance problems at customer sites are frequently caused by the following things:
    - Servers were tested at much less than average or peak workloads. This prevents configuration and tuning problems from being uncovered.
    - Siebel Server components are either incorrectly distributed across servers or are not configured correctly.
    - The load balancing strategy is ineffective under typical workloads. This can be caused by stress testing the servers with a workload that has characteristics different than the production environment.
- **Failover and Resiliency Testing**. Define a test plan that evaluates the effect of server component failures. Undetected single points of failure within the Siebel deployment is a common problem found during implementation readiness reviews by Siebel Expert Services.

  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 under average and peak workloads.
    - **Database server platform failover**. Typically the database server is clustered.
    - **Recovery from database corruption**. Recovery mechanisms are typically provided by the database vendor.
    - **Batch processing support**. Verify that the database server correctly handles batch jobs from servers as well as synchronization requests from Siebel Remote.
    - **Web Client users**. Verify that batch jobs do not degrade transaction processing performance and are completed in a timely fashion.

4 High-Availability Deployment Planning

This chapter includes the following topics:

How Service Failures Affect the Siebel Deployment

This topic describes how major architectural components in a Siebel deployment are affected when a service failure occurs. Services include both hardware platforms and software applications.

Web Clients

Client PC hardware failure and browser crashes are the most common causes of Web Client failure. Operating System crashes can also cause this, but are rare. When the Web Client fails, user sessions will be lost even though the sessions usually continue running on the Siebel Server.

This is because when the 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 will typically need to log in again and start a new user session.

Web Servers

Web servers may fail due to hardware or software issues. Typically, when the Web server fails, Web Clients will not be able to access Siebel Applications, since requests must go through the Web server first. Existing connections from the Web server to Siebel Servers will also be lost.

If Web servers are set up for high availability, for example if there are multiple, load-balanced Web servers, then subsequent requests can be routed to another working Web server. Typically when this occurs, the function of affected Web Client user sessions are not noticeably affected.

Third-Party HTTP Load Balancer for Siebel Servers

Third-party HTTP load balancers handle communication between Web servers and Siebel Servers. Causes of failure differ significantly between hardware-based and software-based solutions. When the load balancer fails, Web Clients and Web servers going through the load balancer will be unable to communicate with Siebel Servers. Network connections in most cases would also be severed, and user sessions will be lost.

If there are multiple, clustered load balancers, then the backup load balancer can take over. Some load balancers can fail over TCP sessions to a backup load balancer. See the vendor's load balancer documentation for details.

When the backup load balancer takes over, user sessions will continue without interruption. However, users sessions will be lost if any of the following occurs:

- A Web Client makes a request while the load balancers are failing over
- TCP sessions are not cleaned up properly on the Web servers

Siebel Servers

Siebel Servers may fail due to hardware or software issues. If the hardware platform fails, or the Siebel Server software fails, then all Siebel Server components will be lost.

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

- **Application Object Managers (AOMs)**. When AOM processes terminate unexpectedly, user sessions hosted by the AOM will be lost. Users must log in to the Siebel application again.

  If users return to the same Siebel Server, SCBroker will attempt to route the user request to a running AOM process.

  If there is only one AOM process and it has failed, then the request will be directed to a different Siebel Server, unless there is only one Siebel Server.

  If AutoStart is enabled, then the Siebel Server process will attempt to restart the terminated AOM process. If successful, the new AOM process will be able to host new user sessions.

- **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 will be returned to the requesting component.
  - **Asynchronous server requests**. An error will be logged but not returned to the requesting component.

  Subsequent requests for the failed batch-mode component will be attempted against either a different instance of the component on the same Siebel Server, or an instance of it on a different server.

  If no instance of the batch-mode component is available, then the request will be 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 Configurator Object Manager, and communication between AOMs and Report Server. Some of these components, such as Report Server and 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, requests to that server will 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 deployment will be 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 may 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 system again.

If the interruption is temporary, interactive server components and most of the batch-mode server components attempt to reconnect with the Siebel Database.

If the interruption is long-term, the Siebel deployment must be shut down and restarted once the database service is restored.

Impact of Service Failures

Table 6 summarizes the impact of failure of services in the Siebel deployment. The table includes information on specific services, not already covered.

Table 6. How Service Failures Affect the Siebel Deployment

Service Failed	Affected Component	Impact
Gateway Name Server	Siebel Server components and Siebel Configurator Object Manager	No new components can be started or added. Users can continue to log in and out of Siebel applications. Existing user sessions are not interrupted. Server requests will continue to be processed successfully. Exceptions are listed below.
	Server administration functions	Unavailable.
	Siebel Reports Server and report functionality	If connection information has been cached, Report Server can still be called. By default, connection information is cached when the connection is made.

Table 6. How Service Failures Affect the Siebel Deployment

Service Failed	Affected Component	Impact
	Siebel Configurator Object Manager	Product configurator sessions can still be launched, as long as the connection information has been cached. By default, the connection information is cached when the first connection is made.
	Name Server database (siebns.dat)	This database maintains server configuration information for the Siebel Enterprise Server. If this database is corrupted or lost, all Siebel Servers must be reinstalled.
Siebel Server	AOM components	Siebel application unavailable. Siebel Connection Broker (SCBroker) failure: New user sessions cannot be created. If a SISNAPI connection is broken between the Web server and the Siebel Server, the user session hosted by the connection will fail. 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).
File System	Attachments	Unavailable.
	Correspondence	Unavailable.
	Shared user preference files	Unavailable.
	Docking transaction files from EIM	Unavailable.
	Email Response	Unable to process inbound messages. Unable to send outbound messages with attachments.
File System Manager (FSM)	Components that access the FSM	Current requests fail.
	Attachments	Unavailable to components that use the FSM.

Table 6. How Service Failures Affect the Siebel Deployment

Service Failed	Affected Component	Impact
Web server	Siebel Web Clients accessing Application Object Managers (AOMs)	Siebel application unavailable to Web Clients. Mobile Web Clients are unaffected.
	EAI inbound HTTP Adaptor	Unavailable.
Siebel Database	Client access, background tasks, batch tasks	Unable to access Siebel eBusiness Applications. The Siebel Enterprise Server cannot function. Only the Mobile Web Client is not immediately affected by a Siebel Database failure.
	Batch and interactive components	Unavailable.

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 that it was connected to, users can reconnect to the system and restart the synchronization process without any data loss.

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

Within a Siebel deployment, a service is defined as one of the following:

- Siebel Gateway Name Server
- Siebel Server
- Siebel Database Server
- Siebel File System
- Web server with the Siebel Web Server Extension (SWSE) installed

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 may not be possible. For example, results of the failure may need to be manually cleaned up. This book does not cover all scenarios, and the customer is 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

Scalable Services (Load Balancing)

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

Application Object Managers (AOMs) are the server components for which load balancing is most frequently provided. When you distribute workload across AOMs, this indirectly distributes workload across the server components that AOMs call. This is called *indirect load balancing*.

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.

Server Clusters

Server clusters consist of two or more physical servers linked together so that should one server fail, 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 (Siebel Gateway Name Server or Siebel Remote) or because of hardware constraints.

Figure 4 illustrates an example of server load balancing and server clustering in a Siebel Enterprise Server

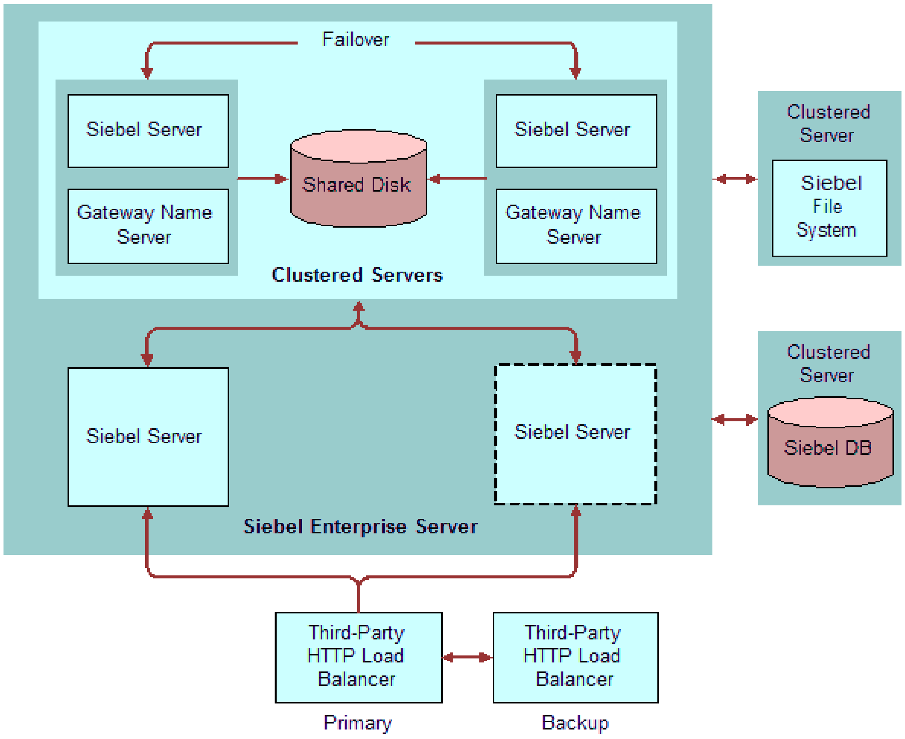

Figure 4. Example of a High-Availability Deployment

Recommended High-Availability Techniques for Specific Services

The three high-availability techniques that Siebel supports are server clustering, load balancing, and resilient processing. Table 7 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 and should be used wherever possible.
- **Supported.** Indicates a high-availability technique is supported for this function. This technique can be used if local conditions prevent using the preferred technique.

- **Blank.** The high-availability techniques in the table are not available for this component.

Table 7. High-Availability Support Matrix for Siebel Components

Component	Clustering	Load Balancing	Resilient Processing
Gateway Name Server database (siebns.dat)	Preferred		
Application Object Managers	Supported	Preferred	
Communications Manager	Supported		Preferred
CORBA Object Manager	Supported		Preferred
Dynamic Assignment	Preferred		
Siebel Configurator	Supported	Preferred. Uses own load balancing method	
eDocument Server	Supported[1]		Preferred
Pricer	Supported		Preferred
EAI (adapters and connectors)	Supported	Preferred, whenever possible[2]	Supported
EAI Object Manager	Supported	Preferred	
Field Service	Supported		Preferred
File System Manager	Supported		Preferred
Interactive Assignment	Supported		Preferred
MQ Series Receiver	Preferred		
Replication Agent	Preferred		
SAP BAPI Integration	Preferred		
SAP IDOC Receiver	Preferred		
SAP IDOC Receiver for MQ	Preferred		
Server Request Broker	Supported		Preferred
Server Request Processor	Supported		Preferred
Siebel File System	Supported		
Siebel Marketing	Supported		Preferred
Siebel Remote	Preferred		
Workflow Monitor	Preferred		
Workflow Process Manager	Supported		Preferred

[1] Supported as long as Microsoft Office has been installed on all clustered nodes. This is particularly helpful in smaller deployments.

2. There are so many different types of EAI deployments that providing a one-size-fits-all recommendation is not practical. For information about the best approach for your deployment, consult Siebel Expert Services.

Best Practices for High-Availability Deployments

Use the best practices below as a starting point for high-availability infrastructure planning.

Profile 1: Global 24X7 Deployment

The deployment has several hundred to tens of thousands of users worldwide requiring 24x7 availability of the Siebel application.

- **Siebel Server load balancers.** A dedicated third-party HTTP load balancer is recommended for this type of deployment. If using hardware load balancers, set up redundant load balancers. Verify that if a load balancer fails, that the remaining load balancers can provide acceptable performance under high workloads.
- **Siebel Gateway Name Server.** Should reside on a dedicated, clustered server pair. Can also reside on Siebel Servers in an existing cluster. Sharing the clustered servers will have 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 has a restriction of one per Siebel Enterprise Server, therefore, load balancing cannot be used.
- **Web servers.** Set up load balancing using one of the standard HTTP load balancers certified by Siebel. Set up the Web server load balancer to allow user requests to fail over to other Web servers. Verify that if a load balancer fails, that the remaining load balancers can provide acceptable performance under high workloads.
- **Siebel Servers hosting an AOM.** Servers hosting an Application Object Manager (AOM) should be load-balanced. Consider AOM or server failure when doing capacity planning. For example, if each Siebel Server can handle 500 users, and you typically have 1500 concurrent users, consider providing four Siebel Servers to handle this load. If one server fails, the other three can still support user loads.

  The Siebel Product Configurator OM is an exception. It includes an internal load balancing mechanism.
- **Siebel Servers hosting other types of components.** Enable batch components on multiple Siebel Servers. Server Request Broker will route requests to these components. This provides resilient processing for batch requests.

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

  - For the AOM 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, data replication techniques such as mirroring and disk arrays should also be used 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

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

- **Load balancers.** If using hardware-based third-party HTTP load balancers, set up redundant load balancers. Verify that if a load balancer fails, that the remaining load balancers can provide acceptable performance under high workloads.
- **Web server.** Set up at least two load-balanced Web servers for high availability.
- **Siebel Servers hosting an AOM.** Either a third-party HTTP load balancer or Siebel load balancing can be used. Third-party HTTP load balancers typically offer more management capabilities, while Siebel load balancing is less complex to setup and maintain. Servers hosting an Application Object Manager (AOM) should be load-balanced. Consider AOM or server failure when doing capacity planning. For example, if each Siebel Server can handle 500 users, and you typically have 1500 concurrent users, consider providing four Siebel Servers to handle this load. If one server fails, the other three can still support user loads.
- **Siebel Servers hosting other types of components.** Same as Profile 1.
- **Siebel Gateway Name Server.** Should reside on a dedicated, clustered server pair and can also reside on Siebel Servers in an existing cluster. Sharing the clustered servers will have minimal performance impact.
- **Siebel File System.** Deploy a clustering technology that has been certified by Siebel Systems. At a minimum, use a RAID 5 disk array for your file system. In addition, make regular backups of your data.
- **Siebel Database.** Deploy a clustering solution supported by your RDBMS vendor. To guarantee data availability and integrity, data replication techniques such as mirroring and the disk arrays should also be used to keep the backup instance of the database in sync with the primary instance.

Profile 3: Limited Resources Deployment

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

Consider collocating multiple Siebel Servers and Web servers on a single computer. Use load balancing for each server type to achieve high availability at minimal cost. Siebel load balancing for the Siebel Servers will work well in this configuration.

To establish high availability, consider putting the Siebel deployment in a two-system cluster. At minimum, make sure that the Siebel Gateway Name Server, Siebel Database server, and Siebel File System are clustered.

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 may use Siebel EAI extensively.

There are no unique high-availability requirements for this profile. Please refer to 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.

If multiple load-balanced Siebel Servers are used, review the recommendations for Profile 2.

5 Load Balancing and Resilient Processing Planning

This chapter includes the following topics:

About Load Balancing

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

Load balancing can be used 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 (AOMs)
- Siebel Configurator (uses own load balancing method)
- Siebel EAI, whenever possible

Prior to Siebel 7.7

Prior to Siebel 7.7, Siebel Systems implemented server load balancing using a third-party software product, Resonate Central Dispatch. When multiple Siebel Servers ran the same Application Object Manager, Central Dispatch distributed server requests across the Siebel Servers. Siebel Servers were integrated with Central Dispatch to maintain session continuity.

Siebel 7.7 and Later

For Siebel 7.7 and later releases, Siebel supports two methods for implementing Siebel Server load balancing:

- Siebel-provided load balancing, called Siebel load balancing. A load balancing module is built into the Siebel Web Server Extension (SWSE). This module provides software-based load balancing for Siebel Servers. Siebel load balancing can be used instead of third-party HTTP load balancers.

- Siebel Systems has certified a number of third-party, hardware-based HTTP load balancers for use in a Siebel deployment. For a list of these, see *System Requirements and Supported Platforms* on Siebel SupportWeb.

  Certification means Siebel Systems has tested interoperability with the load balancer extensively, and specific configuration instructions are available. Siebel Technical Support will work with customers on configuration and interoperability issues specific to Siebel deployments.

  If customers are using a noncertified load balancer and encounter load balancing issues, they should contact the third-party load balancer vendor directly. If a load balancing or connectivity issue is encountered, customers should try to reproduce the issue with Siebel load balancing to isolate the cause. It is recommended that you use a certified load balancer to minimize potential compatibility issues. Although Siebel applications are designed to work with standard, third-party HTTP applications, customers should perform compatibility testing before using an uncertified load balancer.

  Configuration and troubleshooting information for third-party load balancers is available on Siebel SupportWeb as Technical Notes.

Siebel Connection Broker. On each Siebel Server, the Siebel Connection Broker (SCBroker), provides intraserver load balancing. SCBroker distributes server requests across multiple instances of Application Object Managers running on the server.

Resonate support discontinued. Automatic registration of Resonate rules is no longer supported. Siebel Server parameters for Resonate registration are no longer supported. For Siebel Servers, Resonate is classified as an uncertified third-party load balancer.

Figure 5 on page 47 shows an example of third-party load balancing.

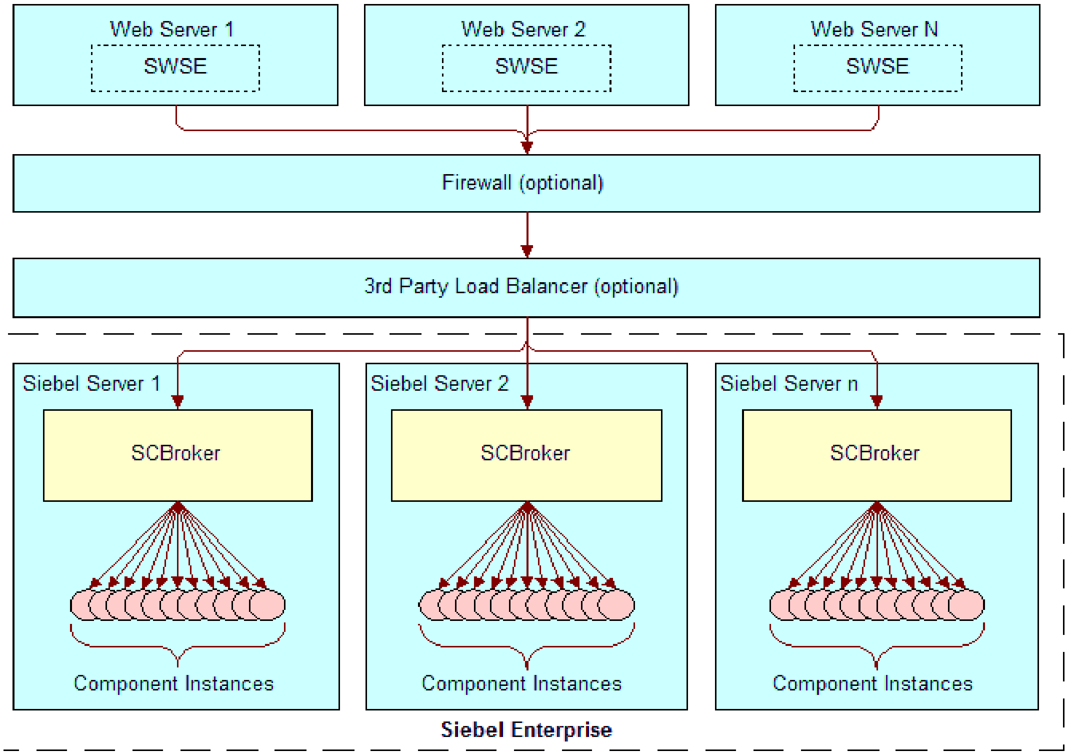

Figure 5. Example of Third-Party Load Balancing

About SISNAPI

Siebel Internet Session Network API (SISNAPI) is a Siebel proprietary message-body format running on top of TCP/IP. SISNAPI is used to communicate between the Web server, Siebel Gateway Name Server, and Siebel Servers. When a client request comes to the Web server, the Siebel Web Server Extension (SWSE) intercepts the request and 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

HTTP Header

When the Siebel Web Server Extension (SWSE) 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. Third-party HTTP load balancers use routing rules to parse the URL and route the message to the correct Siebel Server.

Connection Multiplexing

SISNAPI TCP/IP connections are specific to an Application Object Manager on one Siebel Server. Before opening new connections, the system checks to see if an existing connection is available. If so, the system uses the existing connection. Once the connection is established, it remains open for use by subsequent messages in the session or to be reused by other sessions.

User Request Types

The Siebel Web Server Extension (SWSE) generates three types of user requests. Each causes a new connection to a Siebel Server through the load balancer: initial request, retry request, and reconnect request. The Siebel load balancing module in the SWSE, recognizes these requests types and automatically routes them correctly. If you use a third-party HTTP load balancer, you must configure routing rules to handle these requests:

- **Initial request**. The SWSE generates this request to start a new user session as follows:
  - The SWSE receives the request to start a user session.
  - The SWSE creates the SISNAPI message. The HTTP header in message specifies the Siebel Enterprise Server and desired server component. The message does not specify a Siebel Server name. The SWSE forwards the message to a third-party HTTP load balancer, if installed.
  - The load balancer parses the URL and compares it to routing rules that have been entered in the load balancer.
  - The load balancer uses these routing rules to route the message to a Siebel Server specified in the routing rule. If no SISNAPI connection exists to the Siebel Server, 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 Web server.
  - The Web Server creates a cookie containing the address information. The Web Server receives the cookie information in subsequent session requests. SWSE includes this information in the SISNAPI HTTP header.
  - The load balancer receives subsequent messages and forwards them directly to the specified Siebel Server and server component through the open SISNAPI connection.
- **Retry request**. If a server rejects an initial request, the request is routed back to the SWSE and the following occurs:
  - The SWSE modifies the URL contained in the HTTP header by appending the letters RR to it.
  - The SWSE forwards the message to the load balancer, if installed.

  - The load balancer applies the routing rule that has been entered for RR messages. Typically, this is a round-robin routing rule that forwards the message to another Siebel Server.

- **Reconnect request**. The SWSE generates a reconnect request when it receives a user request for an existing user session that does not have a SISNAPI connection. The SWSE uses the session cookie information to include the server address in the SISNAPI HTTP header.

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

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

About the Load Balancing Configuration File (lbconfig.txt)

The load balancing configuration file provides information about which Siebel Servers will be load-balanced. Its default location is `\siebel\eapps\admin`.

The file has two parts:

- **Session Manager Rules**. The first section contains virtual server definitions used by Siebel load balancing. These definitions map a virtual server name to one or more physical platforms on which Siebel Servers are running. Entries can be edited to create additional virtual servers. Load balancing is managed internally by the load balancing module in the Siebel Web Server Extension (SWSE).

- **Third-Party HTTP Load Balancer Rules**. The second section is provided as a guide for creating routing rules for third-party HTTP load balancers. This section lists a series of Uniform Resource Locators (URLs) that provide a path to Application Object Managers (AOMs). These URLs are included in the HTTP header of SISNAPI messages sent from the Siebel Web Server Extension (SWSE) to the load balancer. They are based on the object manager connect strings located in the Siebel Web Server Extension (SWSE) configuration file (eapps.cfg).

  The entries map these URLs to Siebel Servers where the AOMs are located. The URL and server mapping together can be used to write routing rules for the load balancer. The mapping includes the port number of the SCBroker running on the Siebel Server. SCBroker receives server requests and distributes them to AOMs running on the server.

  These entries are listed in three groups:

  - **Component Rules**. This group lists the servers to use for initial connection requests. The path includes the names of all the servers running the Application Object Manager.
  - **Server Rules**. This group lists the servers to use for server reconnection requests.
  - **Round-Robin Rules**. This group lists the servers to use for retry requests. The URL for these retry requests includes the string RR. There is no significance to the order of the servers in the rule. The third-party HTTP load balancer determines the order in which servers are retried.

For a definition of initial request, reconnection request, and retry request, see "About SISNAPI" on page 47.

lbconfig.txt Session Manager Rules

The syntax of a virtual server definition is as follows:

```
VirtualServer = sid:hostname:SCBroker_port;sid:hostname:SCBroker_port;
```

where:

- `VirtualServer`. The name of the pool of Siebel Servers that will be load-balanced. The default name is VirtualServer. This name is included in the object manager connect strings in the Siebel Web Server Extension (SWSE) configuration file (eapps.cfg). By default, the VirtualServer pool contains all the Siebel Servers running at the time the SWSE was installed. You can edit lbconfig.txt and eapps.cfg to define additional server pools. See "Optimizing the Siebel Load-Balancing Performance" on page 62.
- *sid.* The server ID of the Siebel Server. This is a unique number assigned to each Siebel Server during installation.
- *hostname.* The network host name or IP address of the machine on which the Siebel Server runs. If the machine is part of a cluster, then this should be the cluster virtual host name.
- *SCBroker_port.* The port number of the Siebel Connection Broker.

Third-Party HTTP Load Balancer Rules

The variables in the following rules have the following meaning:

- *enterprise*. The Siebel Enterprise Server name.
- *AOM*. The Application Object Manager name.
- *server*. The Siebel Server name. You can change this to the TCP/IP address of the Siebel Server, if desired.
- *SCBPort*. The port assigned to the Siebel Connection Broker on the Siebel Server.
- *sid.* The server ID of the Siebel Server. This is a unique number assigned to each Siebel Server during installation.

Component Rules

These rules are URLs for initial connection requests. The syntax of a component rule is as follows:

```
/enterprise/AOM/=server:SCBport;...;
```

When the file is generated, a component rule is created for every enabled AOM found on every running Siebel Server.

Server Rules

These rules are URLs for server reconnection requests. The syntax of a server rule is as follows:

/enterprise//!sid.*=server:SCBPort*;

The first asterisk in the syntax is a wildcard for the AOM. The exclamation point and dot-asterisk (.*) are wildcards that parse the server name to extract the server ID.

Not all load balancers can handle a wildcard character (*) in the middle of the URL. In these cases, create URLs with the following format:

/enterprise/AOM/!sid.=server:SCBport*;

Repeat this mapping for each combination of the AOM and Siebel Server ID.

Round-Robin Rules

These rules are URLs for server retry requests. The syntax for a round-robin rule is as follows:

*/enterprise/AOM/*RR=*server:SCBport*;...;

This syntax is the same as that of component rules, except that RR is appended to the URL. This is to alert the load balancer to apply a round-robin rule that routes this request to a different Siebel Server.

Example of a Load Balancing Configuration File

The Siebel Enterprise Server in the example has the following characteristics:

- Enterprise name: Siebel
- Siebel Servers: SiebServA, SiebServB
- Siebel Connection Broker port: 2321 for both servers

Example of lbconfig.txt. Here is an example of an lbconfig.txt file. (Explanatory text at the beginning of the file is not shown):

```
#Section one -- Session Manager Rules:

VirtualServer=1:SiebServA:2321;2:SiebServB:2321;

******************************

#Section two -- 3rd Party Load Balancer Rules

#Component Rules:

/siebel/CRAObjMgr_enu/=SiebServA:2321;SiebServB:2321;

/siebel/eEventsObjMgr_enu/=SiebServA:2321;SiebServB:2321;

/siebel/eMarketObjMgr_enu/=SiebServA:2321;SiebServB:2321;

/siebel/SMObjMgr_enu/=SiebServA:2321;SiebServB:2321;

/siebel/eTrainingObjMgr_enu/=SiebServA:2321;SiebServB:2321;

/siebel/ERMEmbObjMgr_enu/=SiebServA:2321;SiebServB:2321;
```

```
/siebel/ERMAdminObjMgr_enu/=SiebServA:2321;SiebServB:2321;

/siebel/ERMObjMgr_enu/=SiebServA:2321;SiebServB:2321;

/siebel/SalesCEObjMgr_enu/=SiebServA:2321;SiebServB:2321;


#Server Rules:

/siebel/*/!1.*=SiebServA:2321;

/siebel/*/!2.*=SiebServB:2321;


#Round-Robin Rules:

/siebel/CRAObjMgr_enu/RR=SiebServA:2321;SiebServB:2321;

/siebel/eEventsObjMgr_enu/RR=SiebServA:2321;SiebServB:2321;

/siebel/eMarketObjMgr_enu/RR=SiebServA:2321;SiebServB:2321;

/siebel/SMObjMgr_enu/RR=SiebServA:2321;SiebServB:2321;

/siebel/eTrainingObjMgr_enu/RR=SiebServA:2321;SiebServB:2321;

/siebel/ERMEmbObjMgr_enu/RR=SiebServA:2321;SiebServB:2321;

/siebel/ERMAdminObjMgr_enu/RR=SiebServA:2321;SiebServB:2321;

/siebel/ERMObjMgr_enu/RR=SiebServA:2321;SiebServB:2321;

/siebel/SalesCEObjMgr_enu/RR=SiebServA:2321;SiebServB:2321;
```

About the Siebel Connection Broker

The Siebel Connection Broker (SCBroker) is a server component that provides intraserver load balancing. SCBroker distributes server requests across multiple instances of Application Object Managers 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 (AOM) with the least number of running tasks. This creates a user session. Thereafter, SCBroker forwards requests that apply to this session directly to the AOM 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. The port number can be changed.
- **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. Cannot be less than DfltTasks.

- **AutoRestart**. Default is On. If SCBroker terminates abnormally, this allows it to restart automatically. Setting this parameter to Off/False is not recommended.
- **ConnForwardTimeout**. The connection forward time-out determines how long SCBroker will wait for an Object Manager 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, and the request cannot be accepted.

  If a time-out occurs, SCBroker reports an error back to the Web server. The SWSE then modifies the request by appending an RR to the Siebel URL. The SWSE then retries the request by forwarding it to the third-party HTTP load balancer. This causes the load balancer to use the round-robin routing rule to forward the request to the next available Siebel Server.
- **ConnRequestTimeout**. The connection request time-out determines how long SCBroker will wait for all 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, the request is sent back to the Web server in the same fashion as a ConnForwardTimeout.

Choosing a Load Balancing Method

Siebel supports two load balancing methods: Siebel load balancing and third-party HTTP load balancers. For a description of these two methods see "About Load Balancing" on page 45.

Siebel load balancing and third-party HTTP load balancers provide similar features. Table 8 compares key characteristics of Siebel load balancing with those of third-party HTTP load balancers. In the table, SISNAPI is the Siebel protocol used to communicate with Siebel Servers.

Table 8. Load Balancing Method Comparison

Feature Area	Siebel Load Balancing	Third-Party HTTP Load Balancer
Product form	Part of the Siebel Web Server Extension software	Can be a dedicated device or part of an intelligent network switch. If software-based, usually installed on an available server. Considered part of customer's networking infrastructure environment.
Installation	Part of Siebel installation process	Varies by vendor. Hardware-based load balancers must be physically installed on the network. May have network topology restrictions.

Table 8. Load Balancing Method Comparison

Feature Area	Siebel Load Balancing	Third-Party HTTP Load Balancer
Configuration	Supports SISNAPI protocol.	Must define server rules to support routing of SISNAPI connections. Hardware-based load balancers are typically administered using a Web browser. Software-based load balancers provide administration software.
Load balancing scheme	Round-robin only.	Response-time-based, resources-based, or round-robin.
Scalability	No application-imposed hard limit.	Varies by vendor. Typical limiting factors are network traffic throughput and number of servers per load balancing pool.
Server health checks	Connection success/fail is monitored through SWSE stat page. No active checks.	Supports ICMP, TCP, and HTTP health-checks. HTTP health-checks are recommended.
Security & network access	Web server must directly connect to the application server.	Generally supports NAT, VIPs, VPorts. Also supports packet inspection and filtering.
Admin and configuration	Configured using text file. Administered through Siebel Server Administration.	Generally configured and administered through Web interface and command line tools.
Deployment limitations	All load-balanced servers should have same configuration and equal load capacity.	No limitations on load balancer except network topology requirements.

General guidelines. Third-party HTTP load balancers are a good choice when any of the following is true:

- Hardware load balancers are already in use or are preferred.
- They provide desired security features.
- A more sophisticated load-balancing scheme is desired.
- The site requires centralized monitoring and management of system hardware and network infrastructure.

Siebel Load balancing distributes user login requests in a round robin fashion, which works best if all servers are configured equally and have similar capacities:

- Configure all load-balanced Siebel Servers with the same Maximum Tasks setting for an application.

- All load-balanced Siebel Servers should be allocated an equal amount of server resources, such as CPU and memory configuration. For example, you will run Call Center on two Siebel Servers. One of them also will run Field Service. Call Center must compete for resources with Field Services on one of the servers. This is not recommended.

About Setting Siebel Server Task Loads

It is critically important not to set the maximum number of tasks for an Application Object Manager (AOM) or other load-balanced component higher than the server can reasonably handle. The server component parameter MaxTasks, sets the maximum number of sessions an AOM or AOM thread can handle. Use this parameter to set an AOMs maximum workload ceiling.

For example, a Siebel Server can handle 400 tasks for an AOM without degraded performance, but you set the maximum tasks to 500. When the AOM reaches 400 tasks, new tasks continue to be accepted, but server performance begins to degrade noticeably. Load balancing becomes inefficient as tasks are sent to the over-tasked server.

For multithreaded components, MaxTasks specifies the number of tasks that can run within each thread. MaxMTServers specifies the number of threads that the component can run. The total number of tasks that a multithreaded component can run is MaxTasks multiplied by MaxMTServers.

For more information on MaxTasks and MaxMTServers, see *Siebel System Administration Guide* and the *Performance Tuning Guide*.

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, another 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 workload. Resilient processing is about providing redundancy. Resiliency also provides round-robin distribution of workload 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, resilient processing should be used instead of server clustering.

If server clustering is used, resilient processing is not required, as the cluster will provide a resilient environment. Resilient processing is not available for all server components. It cannot be used for server components that must run on only one server. An example is Siebel Remote. In these cases, the only high availability option is server clustering.

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

- Communications Manager
- CORBA Object Manager

- eDocument Server
- Pricer
- Field Service
- File System Manager
- Interactive Assignment
- Server Request Broker
- Server Request Processor
- Siebel Marketing
- Workflow Process Manager

Resilient processing uses two server components:

- **Server Request Processor**. This component handles asynchronous server requests. See "About Server Request Processor" on page 56.
- **Server Request Broker**. This component handles synchronous server requests. See "About Server Request Broker" on page 57.

About Server Request Processor

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

The Server Request Processor runs by default on all 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 the following criteria:

- Be the correct state (Submitted)
- Its start time must have passed
- The target Siebel Server must either not be specified, or must be the Siebel Server on which the SRProc instance is running

If a request is eligible, SRProc will invoke Server Request Broker (SRBroker) to run the request. Therefore, as long as a target Siebel Server is not specified, asynchronous requests will be read by any SRProc task on any Siebel Server.

This provides resilient processing for server-initiated tasks. As long as a SRProc task is running somewhere in the Siebel Enterprise Server, the request will be processed. For more information on resilient processing, see "About Resilient Processing" on page 55.

About Server Request Broker

Server Request Broker (SRBroker) processes synchronous server requests—requests that must be run immediately, and which the calling process waits for completion. Server Request Broker (SRBroker) in Siebel 7 replaces Server Request Manager (SRMSynch) in previous versions.

Whereas SRMSynch could only run server requests on a local Siebel Server (that is, the same Siebel Server that SRMSynch was running on), SRBroker can run server requests on any server in the Enterprise. For example, if SRBroker is unable to run a server request on the local Siebel Server because the required component is not enabled, SRBroker will locate another Siebel Server that is hosting the required component and run it there. Server Request Broker runs by default on all Siebel Servers.

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

- If the required component is available locally then SRBroker will run the task locally.
- If the required component is not available locally then SRBroker will identify any Siebel Servers in the same Enterprise that have the component online. Server requests will be 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 will fail.

This provides 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 on resilient processing, see "About Resilient Processing" on page 55.

6 Managing Siebel Load Balancing

This chapter includes the following topics:

Generating the Load Balancing Configuration File (lbconfig.txt)

You must generate a load balancing configuration file (lbconfig.txt) in the following situations:

- Before installing the Siebel Web Server Extension (SWSE) when you are using Siebel load balancing.
- To provide URLs for routing rules as part of configuring a third-party HTTP load balancer.
- When you add or remove a Siebel Server and you are using either Siebel load balancing or a third-party load balancer.

The load balancing configuration file provides virtual server definitions for Siebel load balancing. It also provides URLs for writing connection rules for third-party HTTP load balancers.

For information on the syntax of the load balancing configuration file, see "About the Load Balancing Configuration File (lbconfig.txt)" on page 49.

Prerequisites. Generating the file has the following prerequisites:

- Verify that all the Siebel Servers for which you want to provide load balancing are running.
- On each Siebel Server, verify that the Application Object Managers (AOMs) you want to load balance are enabled. Disable any AOMs that will not be used.

NOTE: If you have optimized the existing lbconfig.txt by creating multiple virtual server definitions, you will lose these changes when you generate the file. To prevent this, save the file under another name before generating it. Then copy your changes to the new file. For information on optimizing lbconfig.text, see "Optimizing the Siebel Load-Balancing Performance" on page 62.

To generate the lbconfig.txt file

1. On a Siebel Server, start the Server Manager at the enterprise level (do not use the /s option) and enter the following command:

   ```
   generate lbconfig
   ```

   This generates the lbconfig.txt file. The file is stored in the admin subdirectory of the Siebel Server installation directory.

2. Copy the lbconfig.txt file to the Web Server directory on all Web servers that will be used for load balancing.

   As an alternative, copy the file to a shared filesystem location accessible by all the Web servers.

   When installing the Siebel Web Server Extension on a Web server, you will be prompted for the location of this file.

3. Restart the Web server.

Manually Enabling Siebel Load Balancing

When you install the Siebel Web Server Extension (SWSE), the installation wizard asks if you want to enable Siebel load balancing. The installation wizard then adds Siebel load balancing information to the SWSE configuration file (eapps.cfg).

If you want to manually enable or disable Siebel load balancing, or you have changed the location of the load balancing configuration file (lbconfig.txt), you must edit the eapps.cfg file to reflect these changes.

To edit Siebel load balancing entries in the eapps.cfg file

1. Go to the SWSE installation directory and locate the `bin` subdirectory.
2. Using a text editor, open the SWSE configuration file, `eapps.cfg file`.

3 Locate the `[ConnMgmt]` section and edit the variables as shown in Table 9.

Table 9. Configuration Variables in ConnMgmt Section of eapps.cfg

Variable Name	Acceptable Values	Description
EnableVirtualHosts	True or False	■ Set to TRUE to enable Siebel load balancing. ■ Set to FALSE to disable Siebel load balancing. If configuring a third-party HTTP load balancer, this variable must be set to FALSE.
VirtualHostsFile	`<pathname>`	Enter the full path to the lbconfig.txt file. The default location: `\siebel\eapps\admin\lbconfig.txt` If you have multiple Web servers, consider placing the lbconfig.txt file on a shared drive accessible by all the Web servers.

4 Save the file.

5 Restart the Web server.

6 Repeat these steps for all Web Servers on which the Siebel Web Server Extension is installed.

Changing the Enterprise Configuration Under Siebel Load Balancing

The most common configuration changes that affect load balancing performance are as follows:

- Adding or removing Siebel Servers
- Enabling or disabling Application Object Managers (AOMs)

These changes require that you edit the load balancing configuration file (lbconfig.txt). For information on the function and layout of the lbconfig.txt file, see "About the Load Balancing Configuration File (lbconfig.txt)" on page 49.

Adding or Removing Siebel Servers

If you add or remove Siebel Servers that are being load-balanced, you must revise the load balancing configuration file (lbconfig.txt) to add or remove the servers from the VirtualServer definition. After you revise the file, restart the Web server.

Do this for all Web servers on which the Siebel Web Server Extension (SWSE) is installed. You do not need to revise the SWSE configuration file (eapps.cfg).

Use one of the following methods to revise the file:

- The recommended method for revising the lbconfig.txt file is to regenerate it. See "Generating the Load Balancing Configuration File (lbconfig.txt)" on page 59.
- If you have optimized the file, consider editing the file instead. This preserves your existing changes. See "Optimizing the Siebel Load-Balancing Performance" on page 62.

Enabling or Disabling Application Object Managers (AOMs)

If you enable or disable a load-balanced Application Object Manager, you must edit the load balancing configuration file (lbconfig.txt) if either of the following is true:

- You are enabling an AOM on a Siebel Server that is not included in the VirtualServer definition in lbconfig.txt. Edit the definition to add the server.
- You are disabling an AOM on a server, and the AOM is the only one being load-balanced on the server. To prevent failed connection attempts, remove the Siebel Server from the VirtualServer definition in lbconfig.txt.

After you save the file, restart the Web server. Do this for all Web servers on which the Siebel Web Server Extension (SWSE) is installed. You do not need to edit the SWSE configuration file (eapps.cfg).

Optimizing the Siebel Load-Balancing Performance

By default, when the lbconfig.txt file is generated, all Siebel Servers are mapped to a single virtual server. This virtual server name is then added to all Application Object Manager (AOM) connect strings in the Siebel Web Server Extension configuration file (eapps.cfg). This means that the Siebel Web Server Extension (SWSE) will distribute requests for all AOMs to all Siebel Servers.

When requests for an AOM are sent to a Siebel Server on which the AOM is not running, these requests fail. When this occurs, the SWSE automatically sends the failed request to another Siebel Server. Typically, users do not notice these retries unless the allowed maximum number of retries is exceeded.

The allowed maximum number of retries is five. Therefore, if there are more than five load-balanced Siebel Servers on which an AOM is not running, you should consider optimizing the load balancing configuration file. This will prevent users from experiencing failed attempts to start applications.

You optimize lbconfig.txt by adding additional virtual server definitions that define the groups of Siebel Servers on which particular AOMs run. You then edit the AOM connection strings in the SWSE configuration file (eapps.cfg) to include the virtual server specific to that AOM.

For example, you have two Siebel Servers, Sieb1 and Sieb2. They run the AOMs shown in Table 10.

Table 10. AOMs Running on the Siebel Servers

Sieb1	Sieb2
Call Center	Call Center
Sales	Sales
eChannel	Marketing

To minimize retries, you would delete the existing definition, VirtualServer, in lbconfig.txt and define four virtual servers as follows:

```
#Section one -- Session Manager Rules:

CallCenterVirtualServer=1:sieb1:2321;2:sieb2:2321;

SalesVirtualServer=1:sieb1:2321;2:sieb2:2321;

eChannelVirtualServer=1:sieb1:2321;

MarketingVirtualServer=2:sieb2:2321;
```

You would edit the connect strings in the SWSE configuration file (eapps.cfg) to look like this:

```
Call Center: ConnectString = siebel.TCPIP.none.none://CallCenterVirtualServer/
siebel/sscObjMgr_enu

Sales: ConnectString = siebel.TCPIP.none.none://SalesVirtualServer/siebel/
sseObjMgr_enu

eChannel: ConnectString = siebel.TCPIP.none.none://eChannelVirtualServer/siebel/
eChannelObjMgr_enu

Marketing: ConnectString = siebel.TCPIP.none.none://MarketingVirtualServer/siebel/
smeObjMgr_enu
```

For more information on the layout of the lbconfig.txt file, see, "About the Load Balancing Configuration File (lbconfig.txt)" on page 49.

NOTE: If you optimize lbconfig.txt by creating multiple virtual server definitions, you will lose these changes if you generate the file again. To prevent this, save the file under another name before generating it. Then copy your additional virtual server definitions to the new file. For information on generating lbconfig.txt, see "Generating the Load Balancing Configuration File (lbconfig.txt)" on page 59.

To optimize the load balancing configuration file

1. Start Siebel Server Manager and enter the following command to obtain Siebel Server IDs.

   ```
   list server show SBL_SRVR_NAME, SV_SRVRID
   ```

   Write down the server IDs of the servers you want to add to virtual server definitions.

2 Open the lbconfig.txt file with a text editor.

   Its default location is `\siebel\eapps\admin`.

3 In Section One, add additional virtual server definitions. Save the file.

4 Open the SWSE configuration file, eapps.cfg with a text editor.

   Its default location is in *`SWSE_install`*`\admin`, where *`SWSE_install`* is the installation directory for the Siebel Web Server Extension.

5 Change the virtual server name in the desired Application Object Manager connect strings. Save the file.

6 Restart the Web server.

Troubleshooting Siebel Load Balancing

This topic provides guidelines for resolving problems with Siebel load balancing. To resolve a problem, look for it in the list of Symptoms/Error messages in Table 11.

Some problem solutions in the table require changing the function of server components. For information on managing servers and components, see *Siebel System Administration Guide*.

After resolving a problem, if you need to manually rebalance the load between Siebel Servers, see "Manually Rebalancing Siebel Server Loads" on page 69.

Table 11. Resolving Siebel Load Balancing Problems

Symptom/ Error Message	Diagnostic Steps/ Cause	Solution
Users do not get a login page. Browser may display "Server Busy Error."	**1** Verify IP access to Siebel Servers.	See "Verifying IP Access to Siebel Servers" on page 68.
	2 Verify TCP port access on Siebel Servers.	See "Verifying Load Balancing Port Access for Siebel Servers" on page 68.
	3 Verify that the SWSE is configured correctly.	The SWSE configuration file (eapps.cfg) is located in `*SWSE_install_dir*\bin`. Open the file and check the following: ■ EnableVirtualHosts=True ■ VirtualHostFile is set to the full path to the load balancing configuration file (lbconfig.txt). The default location for this file is as follows: `\siebel\eapps\admin` ■ For each load-balanced Application Object Manager, verify that the virtual server specified in the connect string matches the one in lbconfig.txt. See "About the Load Balancing Configuration File (lbconfig.txt)" on page 49.

Table 11. Resolving Siebel Load Balancing Problems

Symptom/ Error Message	Diagnostic Steps/ Cause	Solution
	4 Verify that Siebel load balancing is configured correctly.	The default location for the Siebel load balancing configuration file (lbconfig.txt) is `\siebel\eapps\admin` Typically, this file is generated automatically. If you have edited the virtual server definition, do the following: ■ Verify that the syntax of the virtual server definition is correct. See "About the Load Balancing Configuration File (lbconfig.txt)" on page 49 ■ For each Siebel Server in a virtual server definition, verify that the server ID (sid) is correct. See "Optimizing the Siebel Load-Balancing Performance" on page 62.
	5 Check if a Siebel Server has been reinstalled or reconfigured.	If so, the Siebel load balancing configuration file (lbconfig.txt) must be edited or regenerated. See "Generating the Load Balancing Configuration File (lbconfig.txt)" on page 59.

	6 Increase the SWSE logging level.	To turn on detailed SWSE logging, set the following environment variables:
		`SIEBEL_SESSMGR_TRACE=1`
		`SIEBEL_LOG_EVENTS=ALL`
		Then restart the Web server.
		If this logging level does not reveal the problem, set
		`SIEBEL_SISNAPI_TRACE=1.`
		This greatly increases the logging level for SISNAPI message handling.

	7 Configure a Web Server to connect directly to a Siebel Server.	Open the SWSE configuration file (eapps.cfg) and edit the connect string for an Application Object Manager to specify a known good Siebel Server. Restart the Web server and try to log in. If the login succeeds, then the problem is with the Siebel load balancing configuration. If the login fails, the problem is related to network connectivity.

Table 11. Resolving Siebel Load Balancing Problems

Symptom/ Error Message	Diagnostic Steps/ Cause	Solution
Users can connect but loads are not balanced evenly between Siebel Servers	■ Unequal loads may be caused by characteristics of users and jobs.	Because jobs are distributed in a round-robin fashion, it is normal for a snapshot of the servers to show somewhat unequal loads. This can be caused by several things, including the nature of the jobs and the rate at which users log in and log out on different servers. Over a longer period, the number of sessions handled by each server will even out.
	■ Siebel Servers do not have equal access to computing resources.	Verify that all Siebel Servers have equal access to computing resources such as CPU and memory.
	■ A Siebel Server has recently added or has been restarted.	Load balancing is based on user logins. As current sessions are terminated and new sessions started, the new Siebel Server will be included in the load sharing.
	■ A Web server cannot route requests to one or more Siebel Servers.	Check for connectivity problems between the Web servers and the Siebel Server with the low workload as described earlier in this table.
	■ A Siebel Server is rejecting an unusual number of user requests.	Check the SWSE log files for "SISNAPI Connection Refused" messages. Possible causes are: 1 SCBroker is either not running or is listening on the wrong port. 2 The requested Application Object Manager is not running or cannot run any more tasks. 3 The requested Application Object Manager has a hung task or thread. 4 The Application Object Manager cannot communicate with the database server.
	A Siebel Server has functional or configuration problems.	Enable server diagnostics. Look for problems with components. Verify basic configuration is correct.

Verifying IP Access to Siebel Servers

This task is part of "Troubleshooting Siebel Load Balancing" on page 65.

To verify IP access to Siebel Servers

1. Open the lbconfig.txt file.

   Its default location is `\siebel\eapps\admin`.

2. Write down the exact string used to identify the Siebel Servers in the Virtual Server definitions.

   This string will either be a host name or an IP address.

3. On the Web servers where SWSE is running, ping each Siebel Server. Use the string from the lbconfig.txt file.

   If the ping succeeds then there is IP access.

4. If the ping does not succeed, complete the remaining steps that follow.
5. Verify that the Siebel Servers are on the network and running.
6. Check for basic networking problems such as cabling, routers, and so on. Verify there is a physical path between the Web Servers and Siebel Servers.
7. If the Siebel Servers are part of multiple networks, verify that the Web servers and Siebel Servers have a network in common.
8. If you used the host name to do the ping, verify that the Siebel Servers are registered correctly in the DNS and that the names resolve to the correct IP address.
9. Verify that a networking device such as a router or firewall are not blocking access to the Siebel Servers.

Verifying Load Balancing Port Access for Siebel Servers

This task is part of "Troubleshooting Siebel Load Balancing" on page 65.

To verify load balancing port access

1. On the Web servers where SWSE is running, telnet to the SCBroker port (2321) on each Siebel Server.

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

   ```
   telnet SiebSrvr1:2321
   ```

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

2. If the connection fails, "Could not open connection to server," then complete the remaining steps that follow.
3. Verify that the desired Siebel applications are running on each Siebel Server.

4. On each Siebel Server, verify that SCBroker is running and is configured to listen on port 2321.
5. Verify that the operating system is not blocking access to the SCBroker port.
6. Check that no other networking device, such as a firewall, is blocking access to the SCBroker port.

Manually Rebalancing Siebel Server Loads

Server loads can become unevenly distributed for several reasons:

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

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

Whenever possible, let normal user login behavior rebalance Siebel Server workloads. Manually intervene only when absolutely necessary. Use one of the following methods to manually rebalance server workloads:

- Stop SCBroker on a Siebel Server. This directs workload away from that server. This does not impact existing user sessions. However, SISNAPI session reconnect does not work for this server. If the SISNAPI connection times out, and user requests are coming through a Web server other than the one used for log in, the session will be lost.
- Modify the Siebel load balancing configuration file (lbconfig.txt) to remove a Siebel Server. Then restart the Web server. This removes the Siebel Server from load balancing and directs its workload to other servers. If you have only one Web server, this terminates all user sessions. If you have multiple Web servers, users making a session request may experience session termination. Use this method only as a last resort.

7 Managing Third-Party Load Balancing

This chapter includes the following topics:

Setting Up Third-Party HTTP Load Balancers

Third-Party load balancers receive SISNAPI messages from the Web server. The load balancer routes these messages based on the URL that they contain. To configure an HTTP load balancer, you must write connection rules that route these messages to the correct Siebel Servers.

Siebel Systems provides a utility to generate these rules. The utility reviews the configuration of the Siebel Servers. It then generates a file that pairs connection strings included in SISNAPI messages with paths to the correct Siebel Servers. The rules are stored in the load balancing configuration file (lbconfig.txt). Use this file to help configure the load balancer.

The file provides three types of connection rules: component rules, server rules, and round-robin rules. These rules types are mandatory. You must include all three types when you configure the load balancer.

- Not configuring round-robin rules can cause login failures.
- Not configuring server rules can cause unexpected session termination.

For a full explanation of the configuration file, see "About the Load Balancing Configuration File (lbconfig.txt)" on page 49.

For information on SISNAPI, see "About SISNAPI" on page 47.

Most load balancers allow you to associate a virtual IP (VIP) address and port number with a group of load balancing rules. They also allow you to define servers as resources and to create groups for them. The procedure below outlines the general steps for setting up load balancers for Siebel Servers.

Prerequisites

- The third-party HTTP load balancer should be one certified by Siebel Systems. For a list of these load balancers, see *System Requirements and Supported Platforms* on Siebel SupportWeb. Noncertified load balancers must have the following characteristics:
  - Must be an HTTP load balancer capable of level 7 HTTP routing. Must be able to parse URLs in HTTP headers.

  - Must allow end-points to manage TCP connections. Specifically, must allow one-to-one mapping between client and server TCP sessions. Also, must not do back-end connection pooling, such as reverse proxy server pooling.

- Verify that all the Siebel Servers for which you want to provide load balancing are running.
- On each Siebel Server, verify that the Application Object Managers (AOMs) you want to load balance are enabled. Disable any AOMs that will not be used.
- Prior to installing the Siebel Web Server Extension, select an unallocated, static VIP address and port number for the load balancer.
- Generate the load-balancing configuration file (lbconfig.txt). To create the file, see "Generating the Load Balancing Configuration File (lbconfig.txt)" on page 59. Review the HTTP load balancer rule types: component rules, server rules, and round-robin rules.
- Install the Siebel Web Server Extension on the desired Web servers. The installation wizard will ask you to choose Siebel load balancing or third-party load balancing. Choose third-party load balancing and enter the VIP address and port number for the load balancer.
- At least one Siebel Server must be installed and running.

To set up a third-party HTTP load balancer

1. Install and complete initial configuration of the third-party HTTP load balancer.

   Refer to the vendor documentation for details.

2. Verify that the load balancer can work with the machines that will host the Siebel Servers.

   Refer to the vendor documentation for networking requirements.

3. Add the desired Siebel Servers to the load balancer as pools of resources.

   Typically, each resource is defined as a combination of hostname or IP address, and TCP Port. For Siebel Server load balancing, use the hostname or IP address of the Siebel Server, and the SCBroker port.

4. Create load balancing rules or content rules in the load balancer.

   Load Balancing rules are mappings between URLs and pools of resources. For each line in the lbconfig.txt file, create one such mapping or rule in the load balancer.

   Check Siebel SupportWeb for specific configuration instructions and automatic configuration scripts for certified load balancers.

   NOTE: You must configure the HTTP load balancer to handle all three types of rules: component, server, and round-robin.

5. For each group of load balancing rules, define the desired load balancing scheme.

   For component rules, use any preferred load balancing scheme.

   For server and round-robin rules, a round-robin load balancing scheme is recommended.

6. Define a VIP address and virtual port for all the load balancing rules.

   The VIP and virtual port must match the VIP and virtual port specified in the object manager connect strings of the Siebel Web Server Extension configuration file (eapps.cfg).

   This file is located in *SWSE_install*\bin, where *SWSE_install* is the Siebel Web Server Extension installation directory.

7. If the load balancer has a configurable TCP connection time-out, adjust the time-out so that it is greater than the SISNAPI ConnIdleTime setting.

   This prevents the load balancer from disconnecting active SISNAPI sessions. For information on setting the SISNAPI ConnIdleTime, see the *Siebel System Administration Guide*.

Revising the Third-Party HTTP Load Balancer Configuration

You must revise the third-party HTTP load balancer configuration or edit the Siebel Web Server Extension file (eapps.cfg) if you do either of the following:

- Add or remove a Siebel Server that is load-balanced.
- Enable or disable an Application Object Manager that is load-balanced.

Prerequisites

- Verify that all the Siebel Servers you want to load-balance are running.
- Verify that the Application Object Managers (AOMs) you want to load-balance are running. Disable any AOMs you do not want to load balance.
- Obtain the virtual IP (VIP) address and port number for the load balancer.
- Review the layout of the load balancing configuration file. See, "About the Load Balancing Configuration File (lbconfig.txt)" on page 49.

Several of the steps in the following procedures are about manually modifying the configuration of the load balancer. If a script is available that automatically imports server configurations, run this script instead.

To add or remove a Siebel Server

1. Run the SWSE configuration wizard.

   See the *Siebel Installation Guide* for the operating system you are using.

   This updates the Siebel Web Server Extension configuration file (eapps.cfg) to reflect the new or removed server.

2. Create a new load balancing configuration file (lbconfig.txt).

   See "Generating the Load Balancing Configuration File (lbconfig.txt)" on page 59. This updates the URL mappings in the file to reflect the new or removed server.

3 Use a text editor to view the load balancing configuration file (lbconfig.txt).

   See "About the Load Balancing Configuration File (lbconfig.txt)" on page 49. Refer to the file to obtain URLs for editing rules in the steps below.

4 Start the load balancer configuration software.

5 Update the resource group definitions to reflect the added or removed server.

6 Revise the component and round-robin rules to reflect the added or removed Application Object Manager (AOM) running on the server.

7 If adding a server, create a server rule. If deleting a server, delete the server rule.

8 Save the configuration.

To add or remove an Application Object Manager on a Siebel Server

1 Run the SWSE configuration wizard.

   See the *Siebel Installation Guide* for the operating system you are using.

   This updates the Siebel Web Server Extension configuration file (eapps.cfg) to reflect the new or removed (disabled) Application Object Manager.

2 Use a text editor to view the load balancing configuration file (lbconfig.txt).

   See "About the Load Balancing Configuration File (lbconfig.txt)" on page 49. Refer to the file to obtain URLs for editing rules in the steps below.

3 Start the load balancer configuration software.

4 Revise the component and round-robin rules to reflect the added or removed Application Object Manager (AOM).

5 If adding a new AOM with a new VIP, edit the object manager connect string for the AOM in the Siebel Web Server Extension configuration file (eapps.cfg).

   The default location of the configuration file is `\bin\eapps.cfg` in the Siebel Web Server Extension installation directory.

6 Save the configuration.

No changes are required to the server rules that manage reconnection requests in the load balancer.

Setting the Load Balancer Connection Time Out

Many third-party HTTP load balancers allow you to set a connection time-out. When the time-out occurs, the SISNAPI connection to the application object manager (AOM) on the Siebel Server is terminated.

In addition, AOMs have a configurable time-out parameter, Connection Maximum Idle Time (ConnIdleTime). When a session is idle for the specified time, the AOM closes the session.

Set the load balancer time-out to be slightly longer than the Connection Maximum Idle Time of the AOMs for which it will provide load balancing.

For example, if the AOM Connection Maximum Idle Time is 600 seconds, set the load balancer connection time-out to 601 seconds or higher.

Avoid setting the AOM Connection Maximum Idle time to be greater than the load balancer connection time-out. This can cause login screen delays and communications performance problems.

Monitoring Servers with Third-Party HTTP Load Balancers

Most third-party HTTP load balancers support server health monitoring. To set up Siebel Server monitoring, configure the load balancer to send an HTTP GET to the server URL. Here is an example URL:

```
//SiebSvr1:2321/siebel/SCBroker
```

- **SiebSvr1.** The Siebel Server host name or IP address
- **2321.** The port number for the Siebel Connection Broker. The default is 2321.
- **siebel**. The Siebel Enterprise Server name
- **SCBroker**. The Siebel Connection Broker

If the Siebel Server and Siebel Connection Broker are running, the Siebel Connection Broker returns the string: `SCBroker OK`. For an overview of the Siebel Connection Broker, see "About the Siebel Connection Broker" on page 52.

This confirms that the Siebel Server is running on the specified platform and that SCBroker is listening at the specified port. This health check does not verify that specific Application Object Managers (AOMs) or other server components are running on the platform.

CAUTION: Do not use TCP Health Check. It may connect to SCBroker and remain connected. This causes SCBroker to wait until the SCBroker component parameter ConnRequestTimeout expires. During this period, SCBroker cannot handle new user-session requests.

Best Practices for Setting Up Monitoring

Implement the following best practices when you set up server monitoring:

- On the Siebel Servers you want to monitor, set the Default Tasks and Maximum Tasks for SCBroker to 2. This provides two instances of SCBroker, which helps prevent monitoring requests from delaying handling of user requests.
- Use HTTP 1.0 to do health checks. It terminates connections to SCBroker quickly.

8 Server Clustering Planning

This chapter includes the following topics:

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. Siebel Systems has certified a variety of third-party vendors to provide server clustering for the Siebel deployment. For a list of vendors and requirements, see *System Requirements and Supported Platforms* on Siebel SupportWeb.

When a clustered Siebel Server fails, all 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, the Communications toolbar is disabled, and users must reconnect and log in to the new server.

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

Active-Active Configuration

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

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

NOTE: Configuring an instance of the Siebel Database server and Siebel Server to fail over to each other is supported, but not recommended.

Potential port conflicts. Some Siebel Server components, such as Siebel Connection Broker (SCBroker), Remote Synch Manager, Handheld Synch, and Siebel Gateway Name Server 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 platforms, each running a Siebel Server. If one platform fails, the other will host 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 platforms, there will be a port conflict after failover.

Capacity planning. Active-Active clusters use all the server platforms continuously. This takes 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.

Where to Use Server Clustering

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

- Siebel Gateway Name Server
- Siebel Servers. Individual server components can be clustered, load-balanced, or both. Some Siebel Application Object Managers do not support or require clustering.
- Siebel File System
- Siebel Database Server. Subject to limitations of third-party RDBMS software.
- Web server on which the Siebel Web Server Extension (SWSE) is installed.

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

- Dynamic Assignment
- Email Agent
- MQ Series Receiver
- Replication agent
- SAP BAPI integration
- SAP IDOC Receiver

- SAP IDOC Receiver for MQ
- Siebel Remote. Make sure that the `DockString` parameter in the remote client configuration file is referencing the virtual server name. This must be configured correctly for remote synchronization to work after failover.
- Workflow Monitor

Server Clustering for Some Components Is Not Supported

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

- Siebel Corba Object Manager
- LDAP/ADSI Directory Server. Vendor may provide built-in replication.
- Universal Queuing Server
- Informatica
- Documents Server (Microsoft server-side integration)
- Hummingbird Search Server
- First Logic Data Quality
- CTI hardware/switch
- Chartworks

Server Clustering and Load Balancing Can Be Used Together

You can set up server clustering and load balancing on the same Siebel Server. For example, you have three Application Object Managers, AOM1, AOM2, and AOM3 running on a Siebel Server. You can set up server clustering for AOM1 and AOM2, and set up load balancing for AOM3.

Best Practices for Server Clustering

The best practices below will help promote failover protection for your system. However, these practices are neither exhaustive nor all-inclusive:

- If you have multiple Siebel Servers running that are not clustered, these should be load-balanced.
- Make clustering the Siebel Database Server a high priority because it is a single point of failure. When clustering the Siebel Database Server, have it already installed and running.

  The Siebel Database Server should be the first server to be clustered.
- Install and configure clustering software on each node to detect failure of that node and to recover and manage all servers as a single system.
- All hardware used should be certified for server clustering by the hardware vendor.

- The Siebel installer allows you to install all servers at once for which you have a license. If you will be operating the Gateway Name Server and Siebel Servers as part of a cluster, you must install and configure the Siebel Gateway Name Server and the Siebel Server individually as separate cluster services.
- On the copy you made of the Cluster Deployment Worksheet (located in the appendices of the *Siebel Installation Guide* for the operating system you are using), fill out the section related to server clustering, so that you can refer to this during installation.

About Third-Party Server Clustering Products

Siebel Systems supports several third-party server cluster products. These products are listed in the *System Requirements and Supported Platforms* on Siebel SupportWeb. Some vendors provide documentation on their Web sites that describes how to install these products in a Siebel deployment. This documentation is listed in Table 12.

Table 12. Vendor Documentation for Installing Cluster Products

Vendor	Product	URL
Hewlett Packard	MC/ServiceGuard Cluster	Contact your Hewlett Packard representative.
IBM	HACMP/E	Send email request to ibmsebcc@us.ibm.com
Microsoft (Doc is not specific to Siebel applications)	■ Windows 2000 Server ■ Windows 2003 Server	■ http://www.microsoft.com/windows2000/en/datacenter/help/ **(For Windows 2000 Server)** ■ http://www.microsoft.com/technet/treeview/default.asp?url=/technet/prodtechnol/windowsserver2003/proddocs/datacenter/default.asp **(For Windows 2003 Server)**
Sun	Sun Cluster Data Service for Siebel	http://docs.sun.com/db/doc/817-1540?q=Siebel
Veritas	Volume Manager	Contact your Veritas representative.

Configuring Oracle Real Application Clusters (RAC)

You can obtain high-availability for Oracle 9i database servers by configuring them in an Oracle Real Application Cluster (RAC). There are several configuration requirements:

- The cluster can contain only two database servers.
- One of the database servers must be configured as active, the other passive (Active/Passive Cluster).

For a full description of Oracle Real Application Clusters and how to configure them, refer to Oracle documentation.

Configuring a Real Application Cluster requires the following steps:

- Setting up the database client
- Setting up the database server
- Setting up the database listener

Setting Up the Database Client

Use this procedure to set up a database client.

To set up the database client

1. Verify that the TNS entry in tnsnames.ora contains the listener addresses for the primary and secondary instances.
2. Set other parameters as follows:
   - LOAD_BALANCE = OFF
   - FAILOVER = ON
   - FAILOVER_MODE, RETRIES = 8
   - FAILOVER_MODE, DELAY = 15

   Here is an example of the configuration:

```
ORAC_AP =
    (DESCRIPTION =
    (LOAD_BALANCE = OFF)
    (FAILOVER = ON)
    (ADDRESS = (PROTOCOL = TCP)(HOST = orac1)(PORT = 3000))
    (ADDRESS = (PROTOCOL = TCP)(HOST = orac2)(PORT = 3000))
    (CONNECT_DATA =
      (SERVICE_NAME = qarac)
      (FAILOVER_MODE =
        (TYPE = select)
        (METHOD = basic)
        (RETRIES = 8)
        (DELAY = 15)
      )
     )
    )
```

Setting Up the Database Servers

Perform this procedure on each of the database servers.

To set up the database server

1. Add the following line to init.ora:

   ```
   active_instance_count = 1
   ```

   This enables the primary/secondary instance feature.

2. Restart the database server.

Setting Up the Database Listener

Use this procedure to set up the database listener.

To set up the database listener

1. Use a text editor to open listener.ora.
2. Delete the sid_list_listener entries:

   ```
   sid_list_listener=
    (sid_desc=
    (oracle_
   home=/private/system/db)
    (sid_name=db1))
   ```

   This forces the database listener to obtain information from dynamic service registration.

3. Verify the modified listener.ora file contains only the following entries:

```
listener=
 (description=
   (address=
   (protocol=tcp)
   (host=db1-server)
   (port=1521)))
```

4. Restart the database listener.

9 Data Integrity and Capacity Planning

This chapter includes the following topics:

Sizing the Database for a Siebel Deployment

As with most client-server applications, the overall performance of Siebel eBusiness Applications is largely dependent on the I/O performance of the database server. To promote optimal I/O performance, it is critical that the tables and indexes in the database be arranged across available disk devices in a manner that evenly distributes the I/O load.

The mechanism for distributing database objects varies by RDBMS, depending on the way in which 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 *System Requirements and Supported Platforms* on Siebel SupportWeb.

In your planning, you will need to allocate space for system storage, rollback or temporary storage space, log files, and other system files, as well as space for Siebel data and indexes. If you allocate too little space for your system, performance will be reduced. If you allocate too much, you will waste disk space.

The space needed by the RDBMS will vary 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 on these requirements.

The space required for Siebel data and indexes will vary depending on what Siebel eBusiness Applications functionality you will implement and the amount and nature of data supporting that functionality.

One database platform per Enterprise Server. The Siebel Servers in a Siebel Enterprise Server can connect to only one database. For example, you cannot configure both an Oracle and DB2 UDB database for use by the same Siebel Enterprise Server.

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

1. Determine the total number and types of users of Siebel eBusiness Applications.

   For example, 500 sales representatives and 75 sales managers.

2. Determine the Siebel eBusiness Applications functionality that you will implement and the entities required to support them.

   Typically, the largest entities are as follows:

   - Accounts
   - Activities
   - Contacts
   - Forecasts
   - Opportunities
   - Service Requests

3. Estimate the average number of entities per user (for example, 100 accounts per sales representative) and calculate an estimated total number of records per entity for your total user base.

4. Using standard sizing procedures for your specific database, calculate the average record size per entity and multiply by the total number of records.

   Typically, these entities span multiple physical tables, all of which must be included in the row size calculation. This will 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.

6. Factor growth rates into your total size calculation.

7. Factor a margin of error into your total size calculation.

Database Table Planning

In most implementations, the Siebel tables listed in Table 13 on page 87 and their corresponding indexes are either the most commonly used, or they can be large in some enterprise deployments. For example, the tables S_EVT_ACT, S_CONTACT, and S_ORG_EXT are large in all enterprise-level deployments of Siebel eBusiness Applications. These tables and indexes should be separated across devices. As a general rule, indexes should be in a different tablespace and, if possible, on different physical devices from the tables on which they are created.

Siebel tablespaces on DB2 UDB should be database-managed tablespaces (DMS) rather than system-managed tablespaces (SMS).

Table 13. Frequently Used and Largest Tables, Enterprise Customers

Table Names	Table Names	Table Names
S_ACCNT_CHRCTR	S_DOCK_TXN_LOGT	S_OPTY_POSTN
S_ACCNT_CO_MSTR	S_DOCK_TXN_SET	S_OPTY_PROD
S_ACCNT_POSTN	S_DOCK_TXN_SETT	S_OPTY_TERR
S_ADDR_ORG	S_ESCL_ACTN_REQ	S_OPTY_POSTN
S_ADDR_PER	S_ESCL_LOG	S_ORG_EXT
S_ASSET	S_ESCL_REQ	S_ORG_TERR
S_CALL_LST_CON	S_EVT_ACT	S_PARTY
S_CON_CHRCTR	S_EXP_ITEM	S_PARTY_PER
S_CON_TERR	S_EXP_RPT	S_PARTY_REL
S_ACCNT_CHRCTR	S_EXP_RPT_APPR	S_PARTY_RPT_REL
S_CRSE_TSTRUN	S_IC_CALC	S_POSTN_CON
S_CRSE_TSTRUN_A	S_IC_CALC_IT	S_PROC_REQ
S_CS_RUN	S_IC_CMPNT_EARN	S_PROD_BASELINE
S_CS_RUN_ANSWR	S_IC_TXN	S_PROD_CONSUME
S_CTLGCAT_PATH	S_IC_TXN_IT	S_PROD_SHIPMENT
S_CYC_CNT_ASSET	S_IC_TXN_POSTN	S_PROD_TARGET
S_DNB_CON_MRC	S_INVC_ITM_DTL	S_QUOTE_ITEM
S_DNB_ORG	S_INVLOC_ROLLUP	S_SRM_REPLY
S_DNB_ORG_SIC	S_INVOICE	S_SRM_REQUEST
S_DNB_UPDATE	S_INVOICE_ITEM	S_SRM_REQ_PARAM
S_DOCK_INIT_ITEM	S_INV_LGR_ENTRY	S_SRV_REQ
S_DOCK_TXN_LOG		

Database Recovery Planning

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

IBM DB2 Recovery Planning

The transaction log should be mirrored to guarantee database recovery in the event of a single device failure. The instance home directory must be mirrored, if resources are available. Hardware or operating system mirroring generally provides the best performance.

Oracle Recovery Planning

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

If your organization does not use RAID storage systems, you should, at a minimum, mirror the redo log, as this is essential when a database goes through crash recovery.

Also, when redo logs are mirrored at the RAID storage system level (usually RAID1 or RAID0+1), there is usually no need to mirror them at the Oracle 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, the Oracle control files should be mirrored as well. Otherwise, you can put the Oracle 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 the Oracle database. Any updates to the control file—for example, the current System Change Number (SCN) or transaction tables—ripple across all members of the control file specification.

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. Tablespaces can be used to place objects on multiple physical containers to promote parallel I/O. Spreading the data and index information across several containers (physical devices) can improve the performance of queries.

IBM DB2 Physical Device Planning

Data and log devices should reside on different disk spindles to reduce contention between random and serial I/O. All DB2 devices should reside 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, 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, all containers should reside on raw UNIX disk partitions, except the containers used for LONG VARCHAR data. Containers for LONG VARCHAR data should reside on the UNIX file system to take advantage of the operating system's buffering capabilities. To make sure that your database will perform well, create one container for each available logical or physical disk device.

Data and log devices should reside on different disk spindles to reduce contention between random and serial I/O. Ideally, all DB2 devices should reside on different disk spindles to minimize I/O contention. When this approach is not possible, spread devices that contain database objects that are often used together across different spindles. These objects include tables, their indexes, and commonly joined tables.

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, use of RAID devices and Microsoft's default setting may suffice. A database administrator must do the requisite sizing calculations to assess the performance requirements beforehand.

Database RAID Array Planning

A redundant array of independent disks (RAID) can provide large amounts of 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 may 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, you must separate the indexes whose names end with _P1 from the tables on which they are created. These tables are heavily used in join operations.

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

Microsoft SQL Server RAID Array Planning

Table 14 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.

Table 14. Microsoft SQL Server Recommended Disk Layout

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.
3–5 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, you should place heavily used tables and their corresponding indexes, such as those listed under "Sizing the Database for a Siebel Deployment" on page 85 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 a discussion of this, see "Database Physical Device Planning" on page 88.

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 will depend 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.

10 Application-Level Deployment Planning

Some Siebel applications can be deployed in more than one fashion. 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.

For additional information on application deployment planning, refer to the *Performance Tuning Guide* and the *Siebel System Administration Guide*.

This chapter includes the following topics:

Session Communications

Siebel Email Response

Siebel Configurator

Siebel Workflow Manager

Siebel Reports Server

Siebel Remote and Batch Job Processing

Session Communications Server Components

Session communications refers to using Communications Server components to enable contact center agents or other users to handle interactive communications work items. For example, Siebel CTI supports this capability, enabling agents to handle voice calls 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) and inbound and outbound communications (such as email).

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 (AOM).** This server component 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 AOM.
- **Server Request Broker (SRBroker).** This server component handles communications between the AOM 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 AOM to CommSessionMgr by way of SRBroker.

  SRBroker is used whether CommSessionMgr runs on the same machine as the AOM, or on a different machine.

Additional Siebel Server Components

You may also be using the following Siebel Server components to manage session communications:

- **Communications Configuration Manager (CommConfigMgr).** This server component may optionally be used to cache communications configuration data.
- **Communications Inbound Receiver (CommInboundRcvr).** Receives and queues inbound work items, and queues them for processing by Communications Inbound Processor. Work items may include email messages (for Siebel Email Response), voice work items that are to be routed using Siebel Universal Queuing (for Siebel CTI), or inbound wireless messages for Siebel Wireless Messaging.
  - 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 phone calls for Siebel CTI or 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)**. Processes inbound work items that were queued by Communications Inbound Receiver.
- **Communications Outbound Manager (CommOutboundMgr).** This server component sends outbound email or other types of messages.

Siebel Product Module

In addition to Siebel CTI or Siebel Email Response, you may be using the following Siebel product modules for session communications:

- **Siebel CTI Connect.** This module consists of CTI middleware, communications driver, and sample communications configuration data. Siebel CTI Connect is based on third-party CTI middleware—Intel NetMerge, formerly Dialogic CT Connect. For Siebel CTI Connect, consult Intel documentation provided on the *Siebel eBusiness Third-Party Bookshelf*.
- **Siebel Universal Queuing.** This module routes communications work items to agents.
- **Siebel Smart Answer.** This module analyzes the content of email and search requests and returns an automatic response or suggests one or more responses to the user for approval.

  Siebel Smart Answer is based on third-party products from Banter. Refer to *Siebel Smart Answer Administration Guide* and consult Banter documentation provided on the *Siebel eBusiness Third-Party Bookshelf*.
- **Siebel eCollaboration.** This module helps agents work with customers directly over Web communications channels.

Session Communications Performance Factors

Depending on your deployment, your agents may be handling phone calls (Siebel CTI), email messages (Siebel Email Response), work items of other communications channels, or a combination of these. Use the following factors to analyze system performance:

- **Inbound calls processed per hour.** 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.** 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).** 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 Server and Siebel Server. Think time is an important factor in overall system load. Estimation of think time should approximate actual user usage.

- **Number of concurrent communications users (agents).** The number of concurrent users of session communications features—typically, contact center agents. This figure will be some percentage of the total number of concurrent users on the AOM.
- **Number of work items**. The average number of inbound and outbound work items per 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 Siebel Universal Queuing, or initiate a large number of work items. Supervisors or other users may be defined as agents but may receive only escalated work items, for example.
- **Volume of customer data.** The total volume of customer data. Data volume affects how quickly data can be retrieved for various purposes, such as to perform lookups for screen pops, route work items, or populate the customer dashboard. In many cases, data volume directly affects response times seen by agents. The volume of data should be realistic and the database needs to be tuned 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 may place limitations on the number of agents that may be served at a single contact center site.
- Integration with ACD queues, predictive dialers, or other modules may affect your configurations, affect network traffic, or have other impacts.
- The capacity of your telephony link (between the ACD switch and the CTI middleware) may affect performance.

Session Communications Deployment Planning

Generally, Siebel Communications Server components for session communications, such as CommSessionMgr, should be run on the same Siebel Server machines as those running AOMs. In some cases, however, you must run CommSessionMgr on a different machine than the AOMs. These options are described in detail below.

CTI middleware generally runs on servers located at each contact center facility.

Running CommSessionMgr on AOM Machines

Generally, Siebel Communications Server components for session communications should be run on the same Siebel Server machines as those running AOMs. Such a topology allows the AOM 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 machines.

Set the Enable Communication parameter to TRUE for all AOMs to which your agents will connect. If you are using load balancing, then all AOMs to which requests are distributed should be configured the same way.

Running CommSessionMgr on Dedicated Machines

Sometimes you *must* run CommSessionMgr on a different machine than the AOM components.

CommSessionMgr must run on the same machine 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 machine with that operating system. (Communications drivers are required to be able to run on one of the supported Siebel Server platforms, as described in *System Requirements and Supported Platforms* on Siebel SupportWeb.)

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.

Key Server Components

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

- **Communications Inbound Receiver (CommInboundRcvr).** Receives and queues inbound work items, and queues them for processing by Communications Inbound Processor. Work items may include email messages (for Siebel Email Response), voice work items that are to be routed using Siebel Universal Queuing (for Siebel CTI), or inbound wireless messages for Siebel Wireless Messaging.
  - 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 phone calls for Siebel CTI or 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)**. 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.

Other Siebel Components or Modules

In addition to Siebel Email Response, you may be using the following Siebel components or modules:

- **Siebel Smart Answer.** This module analyzes the content of email and search requests and returns an automatic response or suggests one or more responses to the user for approval.

  Siebel Smart Answer is based on third-party products from Banter. Refer to *Siebel Smart Answer Administration Guide* and consult Banter documentation provided on *Siebel eBusiness Third-Party Bookshelf*.

- **Siebel Assignment Manager.** This module may be used for routing email messages to agents.
- **Siebel Universal Queuing and session communications components.** If you are using Siebel Universal Queuing to route email work items, then additional session communications components apply. The communications toolbar is enabled in the Siebel application to support accepting new work items.

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, refer to *System Requirements and Supported Platforms* on Siebel SupportWeb.

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.** 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, other usage of the CommOutboundMgr component, or of the email system must also be considered. For example, the Send Email command may be configured to send email through CommOutboundMgr.

- **Volume of customer data.** 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.

  If you are deploying Siebel Smart Answer, you must also consider the size of the knowledge base.

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, or when Siebel Universal Queuing is deployed, session communications performance issues also apply. Using Siebel Smart Answer, especially for auto-response capabilities, reduces the number of agents needed to handle incoming email and reduces corresponding demand on session-related computing resources, such as AOM or CommSessionMgr.

Siebel Email Response Deployment Planning

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

Processing of inbound messages associated with a single response group must be handled on a single machine.

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

CommOutboundMgr and Siebel Smart Answer (Smart Answer Manager) may be run together on a different machine (or machines), as appropriate.

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 will give you more flexibility to distribute processing across multiple servers, and thereby avoid processing bottlenecks.

Siebel Configurator Server Components

Siebel Configurator allows users to interactively configure customizable products at the time of order or quote generation. Siebel Configurator uses a constraint-based solution engine that resides on the Siebel Server. This engine evaluates customer choices and generates product configurations that conform to business rules. Business rules are defined using constraint statements contained in models stored on the Siebel File System.

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

- **Application Object Manager (AOM).** The Siebel Configurator solution engine functions within an AOM, such as Call Center Object Manager (SCCObjMgr) for Siebel Call Center.
- **Siebel Product Configurator Object Manager (eProdCfgObjMgr).** This is an AOM containing the Siebel Configurator solution engine. It can be deployed on a separate Siebel Server from where Siebel Configurator sessions are invoked.
- **Siebel File System.** Stores cached object definitions for customizable product models in the CFGCache directory on the Siebel File System

  For more information about elements of the internal architecture of Siebel Configurator, including Instance Broker (Complex Object Instance Service business service) and Object Broker (Cfg Object Broker business service), refer to *Product Administration Guide*.

Siebel Configurator Performance Factors

Siebel Configurator performance has two aspects:

- **Customizable product loading time.** This is the time elapsed from the moment a user clicks Customize in a quote or order until the user interface for the customizable product has been loaded and displayed to the user.
- **Selection response time.** This is the time elapsed from the moment a selection is made by the user until Siebel Configurator returns a response, such as an update to the customizable product or a conflict message.

The key performance factors that influence these times are as follows:

- **Use of Snapshot Mode caching**. This feature caches customizable product models in memory, which significantly reduces the amount of time required to load customizable products for each new user. This feature is particularly useful for improving performance when a product line has a small number of large, complex customizable products. For more information on Snapshot Mode caching, see *Product Administration Guide*.
- **Number of concurrent configuration users.** The number of concurrent users who access customizable product models. This figure will be some percentage of the total number of concurrent users on the AOM.

  More 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 product models.** The total size and complexity of each customizable product model, 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 product models.** The number of customizable product models accessed by users. It is assumed that each user accesses no more than one customizable product model at one time. A given group of concurrent users may access multiple models, however, each of which must be separately cached.

Siebel Configurator Deployment Planning

There are two major approaches to deploying Siebel Configurator:

- Running Siebel Configurator in an AOM component.
- Running Siebel Configurator on one or more dedicated Siebel Servers. Such servers are sometimes referred to as remote servers, because they are remote to the machine on which AOM is running. In general, this section uses the term dedicated servers.

Running Siebel Configurator in an AOM Component

You can run Siebel Configurator in the AOM component, such as for Siebel Call Center.

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

Running Siebel Configurator on Dedicated Servers

You can run Siebel Configurator on one or more dedicated Siebel Server machines using a server component other than the AOM. This component is Siebel Product Configurator Object Manager (eProdCfgObjMgr).

Possible variations on this deployment strategy include:

- Running one eProdCfgObjMgr component with one AOM component
- Running multiple eProdCfgObjMgr components with one AOM component
- Running one eProdCfgObjMgr component with multiple AOM components

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

Workflow Deployment Planning

Siebel Business Process Designer is a customizable business application that allows you to define, manage, and enforce your business processes. It allows you to design complex workflow processes and automate the enforcement of business policies and procedures. For information on using and administering Siebel Business Process Designer, see *Siebel Business Process Designer Administration Guide.*

The application has the following modules:

- **Workflow Processes**. Allows you to 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.

  The Workflow Process Designer is located in Siebel Tools.

- **Workflow Policies**. Allows you to 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**. Used for defining business object states and state transitions.

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 may invoke workflow processes are cached and reused for subsequent requests. When sizing a system, look at the maximum number of workflow tasks you expect to have active at a given time. This determines the maximum number of Object Manager sessions Siebel applications create.

Starting with Siebel 7.0, Business Integration Manager and Business Integration Batch Manager have been deprecated, so if you were using either one in your business processes you need to replace them with Workflow Process Manager or Workflow Process Batch Manager, respectively.

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, you may want to run Workflow Process Manager on multiple Siebel Server machines. 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), you may also want to run multiple multithreaded processes.

If you are going to run several different types of workflows, you should run each type in a separate process. This 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 per process are controlled through the parameters MaxMTServers (Maximum MT Servers), MinMTServers (Minimum MT Servers), and MaxTasks (Maximum Tasks).

These parameters are per Siebel Server. For example, MaxMTServers refers to how many multithreaded processes to run on each Siebel Server machine. For details, refer to *Siebel System Administration Guide*.

Siebel Reports Server and Firewall Planning

If your network infrastructure includes a demilitarized zone (DMZ), you must enable specific ports on Active Portal and iServer.

This requirement applies in the following circumstances:

- The DMZ is boundaried by an outer firewall and an inner firewall. The outer firewall filters traffic between the internet and the reverse proxy server in the DMZ. The inner firewall filters communications between the reverse proxy server and the Siebel deployment.
- Actuate Active Portal is installed in the DMZ.
- Actuate iServer is installed behind the inner firewall along with the Siebel deployment.

You must enable ports as follows to make sure the Reports Server functions normally:

- On the outer firewall, enable the Actuate HTTP Service Communications port. The default port number is 8700.
- On the inner firewall, activate the iServer port. The default is 8000.
- On the inner firewall, activate the PMD port. The default is 8100.

These port numbers are defaults and can be configured.

Planning Batch Processing When Using Siebel Remote

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

This 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 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.

Conditions That Can Cause Missed Transactions

The following conditions in Assignment Manager can cause increased commit times. This increases the risk that the Transaction Processor wait-time will expire before the commit occurs, and Transaction Processor will not process all the transactions in the transaction log.

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, 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 can 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. This together with the conditions above 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 best-practice recommendations. Experiment with applying them in combination to achieve the best performance while minimizing the risk of exceeding the wait-time.

For additional information on understanding gaps in the transaction log, see the Technical Note: *Siebel Remote Transaction Gaps* on SupportWeb.

Monitor the Transaction Processor Logs

As you apply the best-practices techniques described below, use the Transaction Processor logs to see the result and help you optimize system performance.

Transaction Processor writes 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, investigate the cause. The skipped transactions may not have been routed to mobile users. If so, mobile users may have to re-extract the database.

If skipped transactions occur when no batch jobs are running, the gap is most likely permanent and is caused by failed commits. If the gap is permanent, nothing is lost, and mobile users receive the correct information.

Set a lower BatchSize for Assignment Manager

Setting a lower BatchSize, reduces the number of records processed before each commit and thus reduces the commit times. This reduces the risk of exceeding the wait-time. If performance requires that you increase the BatchSize parameter, do so only after analyzing the number of Assignment Manager threads that you have under average and peak workloads. The lower the number of Assignment Manager threads, the higher you can set the BatchSize parameter.

You can obtain performance statistics on Assignment Manager threads, by raising Assignment Manager's log level. For information on raising the log level, see *Siebel System Administration 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.

11 Siebel Client Deployment Planning

This chapter includes the following topics:

About Standard and High Interactivity Modes

In the Web browser, Siebel applications run in one of two modes: standard interactivity or high interactivity.

High interactivity is designed to provide Siebel applications with a user experience similar to that of traditional GUI-based Windows applications. High interactivity reduces the number of page refreshes, compared to Standard Interactivity—when interacting with the application, browsing through records, and so on. This is made possible by requesting data-only updates from the server. The application thus makes optimal use of network bandwidth.

For example, a high interactivity client does not require a page refresh for creating a new record. A user creates a new record by clicking New. A new row is created in a list dynamically, without a page refresh. The user enters the relevant data, then clicks outside of the record to implicitly commit the change—again, without a page refresh.

Some of the features of the high interactivity framework are:

- Fewer page refreshes.
- Support for client-side scripting.
- Support for implicit commit. This feature enables automatic saving when a user steps off of a new or modified record.
- Other usability features. Such features include:
  - Lists displayed in special applets
  - Drag-and-drop column reordering
  - Drag-and-drop file attachments
  - Keyboard shortcuts
  - Smart controls for calendar, calculator and currency
  - Applet scrollbars.

The high interactivity framework requires the Microsoft Internet Explorer browser running in a Windows environment and uses both ActiveX controls and Java. For a list of supported browser versions, see *System Requirements and Supported Platforms* on Siebel SupportWeb.

Deploying Siebel applications in high interactivity mode requires adhering to strict guidelines regarding the deployed operating system, Internet Explorer version and settings, and Java software environment. For a discussion of which Siebel Applications must be deployed in high-interactivity mode and the related browser requirements, see *Siebel System Requirements and Supported Platforms*.

High-Interactivity Application Deployment Planning

High interactivity applications should be deployed in a way that maximizes local browser performance. For an overview of high interactivity mode, see "About Standard and High Interactivity Modes" on page 103.

Several factors influence this performance.

- **Default column display.** Users can select which columns to display in list applets. Deploy applications with the minimum number of columns set to display. Then allow users to select which columns to add. This improves performance by minimizing the number of Web templates and amount of data required to build views in response to user requests.
- **View layout caching.** Administrators can determine the number of views to be cached by the user's browser. When a view is cached, subsequent visits will cause a user request for only a data update. The Web templates required to build the view are retrieved from the browser cache. They are not rebuilt by the Siebel Server again. This improves response time.
- Set the number of views to be cached as high as practical.
- **User Login.** The first login is generally the most time consuming. The client infrastructure caches the main components of the application on first login. Subsequent logins require far fewer resources. Cached objects remain on the client computer until the cache is cleared or a new version of the application configuration is available.
- **Browser version**. Use the latest supported version of Microsoft Internet Explorer in your application development, testing, and deployment environments. Besides fixes, the latest versions frequently include performance enhancements and improved ActiveX and Java support.

Standard-Interactivity Application Deployment Planning

Standard interactivity applications do not allow users to select which columns to display in a view. They also do not support data-only user requests. All user requests require that the Siebel Server load page templates and rebuild each page for display.

For an overview of standard-interactivity mode, see "About Standard and High Interactivity Modes" on page 103.

To maximize system performance, set the number of columns of data displayed in each view to the minimum needed. Also, when creating or modifying Web page templates, keep template designs as simple as possible.

Standard interactivity applications can be run on several types of Web browsers. Test browser performance for all the types of Siebel clients you will deploy. Investigate all browser settings that affect page display, cookie management, or page caching.

To reduce deployment administration costs, standardize on a browser for all users. Be sure to use the same browser type for development, testing, and production environments.

For a list of browsers that support standard interactivity, see *Siebel System Requirements and Supported Platforms*.

Index