

Oracle9iAS InterConnect Adapter for Siebel 2000

Installation and User's Guide

Release 2 (9.0.2)

February 2002

Part No. A95450-01

Part No. A95450-01

Copyright © 2002 Oracle Corporation. All rights reserved.

The Programs (which include both the software and documentation) contain proprietary information of Oracle Corporation; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent and other intellectual and industrial property laws. Reverse engineering, disassembly or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. Oracle Corporation does not warrant that this document is error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of Oracle Corporation.

If the Programs are delivered to the U.S. Government or anyone licensing or using the programs on behalf of the U.S. Government, the following notice is applicable:

Restricted Rights Notice Programs delivered subject to the DOD FAR Supplement are "commercial computer software" and use, duplication, and disclosure of the Programs, including documentation, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement. Otherwise, Programs delivered subject to the Federal Acquisition Regulations are "restricted computer software" and use, duplication, and disclosure of the Programs shall be subject to the restrictions in FAR 52.227-19, Commercial Computer Software - Restricted Rights (June, 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065.

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy, and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and Oracle Corporation disclaims liability for any damages caused by such use of the Programs.

Oracle is a registered trademark, and Oracle*MetaLink*, Oracle Store, Oracle9i, Oracle9iAS Discoverer, SQL*Plus, and PL/SQL are trademarks or registered trademarks of Oracle Corporation. Other names may be trademarks of their respective owners.

Contents

Send Us Your Comments	vii
Preface.....	ix
Intended Audience	x
Documentation Accessibility	x
Organization.....	x
Related Documentation	xi
Conventions.....	xii
1 Introduction	
What is Siebel?	1-2
Supported Systems.....	1-2
Supported Platforms	1-2
Required Files.....	1-3
Siebel Application Server Configuration.....	1-3
2 Installation and Configuration	
Installing the Siebel Adapter	2-2
Preinstallation Tasks	2-2
Installation Tasks.....	2-2
Siebel Adapter Configuration	2-4
Using the Application Parameter.....	2-5
adapter.ini Initialization Parameter File	2-6
Hub.ini	2-6

Agent Connection Parameters	2-7
Siebel Adapter Parameters	2-11
Starting an the Siebel Adapter	2-12

3 Supported Siebel Interfaces

Siebel Interfaces	3-2
Creating an Implemented Procedure	3-2
Importing Attributes from Siebel	3-5
Creating a Subscribed Event.....	3-8

4 Business Component Relationships

Most Commonly Used Siebel Components.....	4-2
Component Relationships.....	4-2
Account	4-3
Account Parent Components.....	4-4
Account Child Components.....	4-4
Business Address	4-6
Business Address Parent Components.....	4-6
Contact.....	4-6
Contact Parent Components	4-8
Contact Child Components.....	4-9
Internal Product	4-9
Price List.....	4-10
Price List Parent Components	4-11
Price List Child Components.....	4-11
Price List Item.....	4-11
Quote	4-12
Quote Parent Components.....	4-13
Quote Child Components	4-13
Quote Item	4-14
Quote Item Parent Components.....	4-14
Opportunity.....	4-14
Opportunity Parent Components	4-16
Opportunity Child Components.....	4-16
Opportunity Orders	4-17

Opportunity Product.....	4-19
Opportunity Product Parent Components.....	4-19
Opportunity Product Child Components.....	4-20
Order Entry Orders.....	4-20
Other Components With Which Order Entry Interacts.....	4-20
Order Entry Orders Parent Components.....	4-22
Order Entry Orders Child Components.....	4-22
Order Entry Line Items.....	4-23
Order Entry Line Items Parent Components.....	4-23
Order Entry Line Items Child Components.....	4-24
RMA.....	4-24
RMA Parent Components.....	4-25
RMA Child Components.....	4-25
Service Request.....	4-25
Service Request Parent Components.....	4-27
Service Request Child Components.....	4-27
Handling Multi-Valued Fields.....	4-28
Frequently Asked Questions.....	4-30

5 Runtime

Configuration Editor.....	5-2
Log On Siebel.....	5-3
General.....	5-4
Siebel Repository.....	5-5
Login Time Out (in seconds).....	5-5

6 Siebel Adapter Functionality

Control Flow Overview.....	6-2
Design Time.....	6-2
Runtime.....	6-2
Siebel Interfaces.....	6-2
Business Services.....	6-2
Business Objects and Business Components.....	6-3

Unformatted DTYPE_PHONE Values 6-4

Index

Send Us Your Comments

Oracle9iAS InterConnect Adapter for Siebel 2000 Installation and User's Guide, Release 2 (9.0.2)

Part No. A95450-01

Oracle Corporation welcomes your comments and suggestions on the quality and usefulness of this document. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most?

If you find any errors or have any other suggestions for improvement, please indicate the document title and part number, and the chapter, section, and page number (if available). You can send comments to us in the following ways:

- Electronic mail: iasdocs_us@oracle.com
- FAX: 650-506-7407 Attn: Oracle9i Application Server Documentation Manager
- Postal service:
Oracle Corporation
Oracle9i Application Server Documentation
500 Oracle Parkway, M/S 2op3
Redwood Shores, CA 94065
USA

If you would like a reply, please give your name, address, telephone number, and (optionally) electronic mail address.

If you have problems with the software, please contact your local Oracle Support Services.

Preface

This preface contains these topics:

- [Intended Audience](#)
- [Documentation Accessibility](#)
- [Organization](#)
- [Related Documentation](#)
- [Conventions](#)

Intended Audience

This guide is intended for those who perform the following tasks:

- install applications
- maintain applications

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible, with good usability, to the disabled community. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Standards will continue to evolve over time, and Oracle Corporation is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For additional information, visit the Oracle Accessibility Program Web site at

<http://www.oracle.com/accessibility/>

Accessibility of Code Examples in Documentation JAWS, a Windows screen reader, may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, JAWS may not always read a line of text that consists solely of a bracket or brace.

Accessibility of Links to External Web Sites in Documentation This documentation may contain links to Web sites of other companies or organizations that Oracle Corporation does not own or control. Oracle Corporation neither evaluates nor makes any representations regarding the accessibility of these Web sites.

Organization

This document contains:

Chapter 1, "Introduction"

This chapter describes the Siebel adapter and the hardware and software requirements.

Chapter 2, "Installation and Configuration"

This chapter describes installation and configuration for the Siebel adapter.

Chapter 3, "Supported Siebel Interfaces"

This chapter describes the supported interfaces for the Siebel adapter.

Chapter 4, "Business Component Relationships"

This chapter describes component relationships for the Siebel adapter.

Chapter 5, "Runtime"

This chapter provides runtime concepts for the Siebel adapter.

Chapter 6, "Siebel Adapter Functionality"

This chapter describes the functionality of the Siebel adapter.

Related Documentation

For more information, see these Oracle resources:

- *Oracle9iAS InterConnect User Guide* in the Oracle9i Application Server Documentation Library
- *Oracle9i Application Server Installation Guide*
- *Oracle9iAS InterConnect Adapter Configuration Editor User's Guide*

In North America, printed documentation is available for sale in the Oracle Store at

<http://oraclestore.oracle.com/>

Customers in Europe, the Middle East, and Africa (EMEA) can purchase documentation from

<http://www.oraclebookshop.com/>

Other customers can contact their Oracle representative to purchase printed documentation.

To download free release notes, installation documentation, white papers, or other collateral, please visit the Oracle Technology Network (OTN). You must register online before using OTN; registration is free and can be done at

<http://otn.oracle.com/admin/account/membership.html>

If you already have a username and password for OTN, then you can go directly to the documentation section of the OTN Web site at

<http://otn.oracle.com/docs/index.htm>

Conventions

This section describes the conventions used in the text and code examples of this documentation set. It describes:

- [Conventions in Text](#)
- [Conventions in Code Examples](#)
- [Conventions for Microsoft Windows Operating Systems](#)

Conventions in Text

We use various conventions in text to help you more quickly identify special terms. The following table describes those conventions and provides examples of their use.

Convention	Meaning	Example
Bold	Bold typeface indicates terms that are defined in the text or terms that appear in a glossary, or both.	When you specify this clause, you create an index-organized table .
<i>Italics</i>	Italic typeface indicates book titles or emphasis.	<i>Oracle9i Database Concepts</i> Ensure that the recovery catalog and target database do <i>not</i> reside on the same disk.
UPPERCASE monospace (fixed-width) font	Uppercase monospace typeface indicates elements supplied by the system. Such elements include parameters, privileges, datatypes, RMAN keywords, SQL keywords, SQL*Plus or utility commands, packages and methods, as well as system-supplied column names, database objects and structures, usernames, and roles.	You can specify this clause only for a NUMBER column. You can back up the database by using the BACKUP command. Query the TABLE_NAME column in the USER_TABLES data dictionary view. Use the DBMS_STATS.GENERATE_STATS procedure.
lowercase monospace (fixed-width) font	Lowercase monospace typeface indicates executables, filenames, directory names, and sample user-supplied elements. Such elements include computer and database names, net service names, and connect identifiers, as well as user-supplied database objects and structures, column names, packages and classes, usernames and roles, program units, and parameter values. Note: Some programmatic elements use a mixture of UPPERCASE and lowercase. Enter these elements as shown.	Enter sqlplus to open SQL*Plus. The password is specified in the orapwd file. Back up the datafiles and control files in the /disk1/oracle/dbs directory. The department_id, department_name, and location_id columns are in the hr.departments table. Set the QUERY_REWRITE_ENABLED initialization parameter to true. Connect as oe user. The JRepUtil class implements these methods.
lowercase italic monospace (fixed-width) font	Lowercase italic monospace font represents placeholders or variables.	You can specify the <i>parallel_clause</i> . Run <i>Uold_release</i> .SQL where <i>old_release</i> refers to the release you installed prior to upgrading.

Conventions in Code Examples

Code examples illustrate SQL, PL/SQL, SQL*Plus, or other command-line statements. They are displayed in a monospace (fixed-width) font and separated from normal text as shown in this example:

```
SELECT username FROM dba_users WHERE username = 'MIGRATE';
```

The following table describes typographic conventions used in code examples and provides examples of their use.

Convention	Meaning	Example
[]	Brackets enclose one or more optional items. Do not enter the brackets.	DECIMAL (<i>digits</i> [, <i>precision</i>])
{ }	Braces enclose two or more items, one of which is required. Do not enter the braces.	{ENABLE DISABLE}
	A vertical bar represents a choice of two or more options within brackets or braces. Enter one of the options. Do not enter the vertical bar.	{ENABLE DISABLE} [COMPRESS NOCOMPRESS]
...	Horizontal ellipsis points indicate either: <ul style="list-style-type: none"> That we have omitted parts of the code that are not directly related to the example That you can repeat a portion of the code 	CREATE TABLE ... AS <i>subquery</i> ; SELECT <i>col1</i> , <i>col2</i> , ... , <i>coln</i> FROM employees;
.	Vertical ellipsis points indicate that we have omitted several lines of code not directly related to the example.	
Other notation	You must enter symbols other than brackets, braces, vertical bars, and ellipsis points as shown.	acctbal NUMBER(11,2); acct CONSTANT NUMBER(4) := 3;
<i>Italics</i>	Italicized text indicates placeholders or variables for which you must supply particular values.	CONNECT SYSTEM/ <i>system_password</i> DB_NAME = <i>database_name</i>

Convention	Meaning	Example
UPPERCASE	Uppercase typeface indicates elements supplied by the system. We show these terms in uppercase in order to distinguish them from terms you define. Unless terms appear in brackets, enter them in the order and with the spelling shown. However, because these terms are not case sensitive, you can enter them in lowercase.	<pre>SELECT last_name, employee_id FROM employees; SELECT * FROM USER_TABLES; DROP TABLE hr.employees;</pre>
lowercase	<p>Lowercase typeface indicates programmatic elements that you supply. For example, lowercase indicates names of tables, columns, or files.</p> <p>Note: Some programmatic elements use a mixture of UPPERCASE and lowercase. Enter these elements as shown.</p>	<pre>SELECT last_name, employee_id FROM employees; sqlplus hr/hr CREATE USER mjones IDENTIFIED BY ty3MU9;</pre>

Conventions for Microsoft Windows Operating Systems

The following table describes conventions for Microsoft Windows operating systems and provides examples of their use.

Convention	Meaning	Example
Choose Start >	How to start a program.	To start the Oracle Database Configuration Assistant, choose Start > Programs > Oracle - <i>HOME_NAME</i> > Configuration and Migration Tools > Database Configuration Assistant.
File and directory names	File and directory names are not case sensitive. The following special characters are not allowed: left angle bracket (<), right angle bracket (>), colon (:), double quotation marks ("), slash (/), pipe (), and dash (-). The special character backslash (\) is treated as an element separator, even when it appears in quotes. If the file name begins with \\, then Windows assumes it uses the Universal Naming Convention.	<pre>c:\winnt "\"system32 is the same as C:\WINNT\SYSTEM32</pre>

Convention	Meaning	Example
C:\>	<p>Represents the Windows command prompt of the current hard disk drive. The escape character in a command prompt is the caret (^). Your prompt reflects the subdirectory in which you are working. Referred to as the <i>command prompt</i> in this manual.</p>	C:\oracle\oradata>
	<p>The backslash (\) special character is sometimes required as an escape character for the double quotation mark (") special character at the Windows command prompt. Parentheses and the single quotation mark (') do not require an escape character. Refer to your Windows operating system documentation for more information on escape and special characters.</p>	<pre>C:\>exp scott/tiger TABLES=emp QUERY=\ "WHERE job='SALESMAN' and sal<1600\" C:\>imp SYSTEM/password FROMUSER=scott TABLES=(emp, dept)</pre>
HOME_NAME	<p>Represents the Oracle home name. The home name can be up to 16 alphanumeric characters. The only special character allowed in the home name is the underscore.</p>	C:\> net start OracleHOME_ NAME_TNSListener

Convention	Meaning	Example
<i>ORACLE_HOME</i> and <i>ORACLE_BASE</i>	<p>In releases prior to Oracle8i release 8.1.3, when you installed Oracle components, all subdirectories were located under a top level <i>ORACLE_HOME</i> directory that by default used one of the following names:</p> <ul style="list-style-type: none"> ■ C:\orant for Windows NT ■ C:\orawin95 for Windows 95 ■ C:\orawin98 for Windows 98 <p>This release complies with Optimal Flexible Architecture (OFA) guidelines. All subdirectories are not under a top level <i>ORACLE_HOME</i> directory. There is a top level directory called <i>ORACLE_BASE</i> that by default is C:\oracle. If you install Oracle9i release 1 (9.0.1) on a computer with no other Oracle software installed, then the default setting for the first Oracle home directory is C:\oracle\ora90. The Oracle home directory is located directly under <i>ORACLE_BASE</i>.</p> <p>All directory path examples in this guide follow OFA conventions.</p>	Go to the <i>ORACLE_BASE\ORACLE_HOME\rdms\admin</i> directory.

Introduction

Oracle connects to Siebel through the Siebel adapter. This chapter covers the following topics:

- [What is Siebel?](#)
- [Siebel Application Server Configuration](#)

What is Siebel?

Siebel is software application allowing organizations to maintain a single source for customer information. Using Siebel, an organization can better facilitate the sales and marketing forces by concentrating all customer information into one source. This source is accessible using multiple channels. A field sales person can consult the Web, the company call center, or any number of networks to access information needed for a customer.

The Siebel eBusiness Application Solution consists of the Siebel Call Center and the Siebel Sales applications. The Siebel Call center application is a full feature contact center. The call center provides a closed loop, seamless, information flow over multi-channel sales, marketing, and customer service operations. The Siebel Sales application enables field sales organizations to share information across sales teams, manage sales pipelines, and sell collaborately across geographies, time zones, and currencies. Under the hood, all applications use Business Components and Business Services. Business Components are representations of a set of one or more joined tables.

See Also: [Chapter 4, "Business Component Relationships"](#) for information on Business Components

Supported Systems

[Table 1-1](#) lists the system to which the Siebel adapter connects:

Table 1-1 *Siebel supported systems.*

Component Support	Required Software
Siebel Adapter	6.2.1

See Also: *Oracle9i Application Server Installation Guide*, Appendix C for hardware requirements

Supported Platforms

The following platforms support the Siebel adapter:

- Windows NT 4.0 service packs 3 through 6a
- Windows 2000 service packs 1 or 2
- Solaris 2.6

- Solaris 7 (2.7)
- HP-UX B.11.00

Required Files

The Siebel adapter uses the following .jar files at both runtime and design time:

- SBLJAccess.jar—Comprises the Java portion of the Siebel adapter.
- The following files comprise the Siebel Java Data Bean provided on the Siebel installation CD. These four jar files must be copied to the `...\oai\9.0.2\lib\` directory:
 - SiebelTcOM.jar
 - SiebelTcCommon.jar
 - SiebelTC_enu.jar
 - SiebelDataBean.jar

Siebel Application Server Configuration

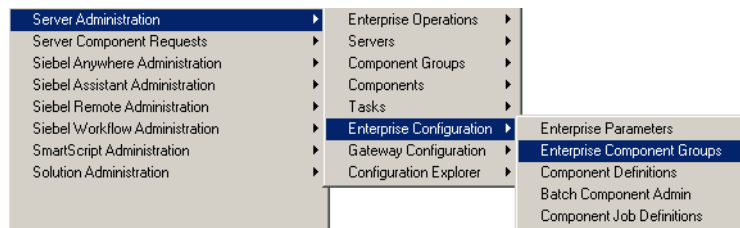
Siebel requires that you enable the Siebel Thin Client Enterprise Component to allow any thin client to communicate remotely with the Siebel Application Server. Ensure that the Siebel Thin Client Enterprise component is enabled on the server.

1. Log on using the Siebel Client (with Server Administration enabled).

The user ID must have sufficient authorization to make server management changes.

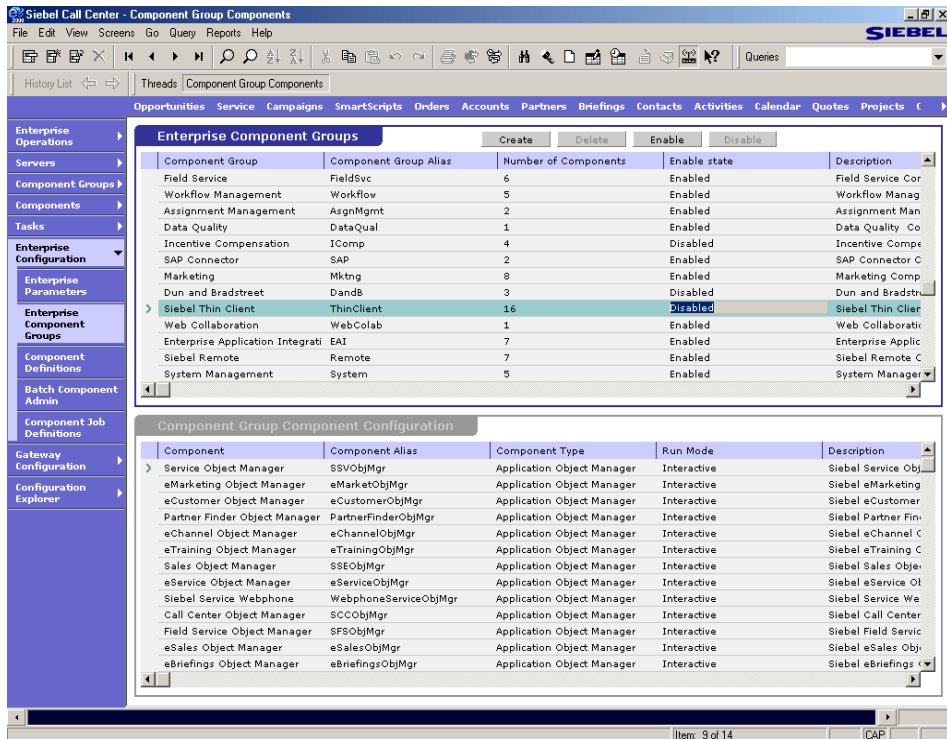
2. Select **Screens->Server Administration->Enterprise Configuration->Enterprise Component Groups**.

Figure 1–1 *Selecting Enterprise Component Groups*



3. Ensure that the **Thin Client** component is enabled.

Figure 1–2 Component Group Components



If it is not enabled, check the component in the list and press **Enable**.

Figure 1–3 Enterprise Component Groups

The screenshot displays the Siebel Call Center interface for managing Enterprise Component Groups. The main window is titled "Enterprise Component Groups" and contains two tables.

Enterprise Component Groups Table:

Component Group	Component Group Alias	Number of Components	Enable state	Description
Field Service	FieldSvc	6	Enabled	Field Service Cor
Workflow Management	Workflow	5	Enabled	Workflow Manag
Assignment Management	AsgnMgmt	2	Enabled	Assignment Man
Data Quality	DataQual	1	Enabled	Data Quality Co
Incentive Compensation	IComp	4	Disabled	Incentive Compe
SAP Connector	SAP	2	Enabled	SAP Connector C
Marketing	Mktng	8	Enabled	Marketing Comp
Dun and Bradstreet	DandB	3	Disabled	Dun and Bradstr
Siebel Thin Client	ThinClient	16	Enabled	Siebel Thin Clie
Web Collaboration	WebColab	1	Enabled	Web Collaboratic
Enterprise Application Integrati	EAI	7	Enabled	Enterprise Applic
Siebel Remote	Remote	7	Enabled	Siebel Remote C
System Management	System	5	Enabled	System Manager

Component Group Component Configuration Table:

Component	Component Alias	Component Type	Run Mode	Description
Service Object Manager	SSVObjMgr	Application Object Manager	Interactive	Siebel Service Obj
eMarketing Object Manager	eMarketObjMgr	Application Object Manager	Interactive	Siebel eMarketing
eCustomer Object Manager	eCustomerObjMgr	Application Object Manager	Interactive	Siebel eCustomer
Partner Finder Object Manager	PartnerFinderObjMgr	Application Object Manager	Interactive	Siebel Partner Fin
eChannel Object Manager	eChannelObjMgr	Application Object Manager	Interactive	Siebel eChannel C
eTraining Object Manager	eTrainingObjMgr	Application Object Manager	Interactive	Siebel eTraining C
Sales Object Manager	SSEObjMgr	Application Object Manager	Interactive	Siebel Sales Obj
eService Object Manager	eServiceObjMgr	Application Object Manager	Interactive	Siebel eService Ol
Siebel Service Webphone	WebphoneServiceObjMgr	Application Object Manager	Interactive	Siebel Service We
Call Center Object Manager	SCCObjMgr	Application Object Manager	Interactive	Siebel Call Center
Field Service Object Manager	SFSObjMgr	Application Object Manager	Interactive	Siebel Field Servic
eSales Object Manager	eSalesObjMgr	Application Object Manager	Interactive	Siebel eSales Obj
eBriefings Object Manager	eBriefingsObjMgr	Application Object Manager	Interactive	Siebel eBriefings

- Restart the machine to ensure that the settings are registered and the component started.

Installation and Configuration

This chapter describes installation and configuration of the Siebel adapter. This chapter discusses the following topics:

- [Installing the Siebel Adapter](#)
- [Siebel Adapter Configuration](#)
- [Starting an the Siebel Adapter](#)

Installing the Siebel Adapter

This section contains these topics:

- [Preinstallation Tasks](#)
- [Installation Tasks](#)

Preinstallation Tasks

The Siebel adapter must be installed in one of the following Oracle homes:

- An existing Oracle9i Application Server Oracle home
- An existing Oracle9i Application Server Infrastructure Database Oracle home
- An existing Oracle9iAS InterConnect Oracle home
- A new Oracle home (the installer creates this for you)

Consult the *Oracle9i Application Server Installation Guide* before proceeding with Siebel adapter installation. This guide includes information on:

- CD-ROM mounting
- Oracle Universal Installer startup
- Oracle9iAS InterConnect software, hardware, and system requirements
- Oracle9iAS InterConnect installation

Note: Oracle9iAS InterConnect Hub is installable through the Oracle9iAS InterConnect Hub installation type. You must install the Oracle9iAS InterConnect Hub before proceeding with Siebel adapter installation.

Installation Tasks

To install the Siebel adapter:

1. Click **Next** on the Welcome page.
The File Locations page displays.
2. Enter the following information in the Destination fields:
 - Name—The Oracle home name.

- Path—The full path to the Oracle home in which to install the Siebel adapter.

Note: Do not change the path specified in the Source field. This is the location on the CD-ROM from which to install the Siebel adapter.

3. Click **Next**.

The Installation Types page displays.

4. Select Oracle9iAS InterConnect Adapters and click **Next**.

The Available Product Components page appears.

5. Select Oracle9iAS InterConnect Siebel Adapter and click **Next**.

6. If the Siebel adapter is not being installed on the same computer as Oracle9iAS InterConnect Hub and another adapter is not installed in the current Oracle home, the Oracle9iAS InterConnect Hub Database page appears. Enter the following information about the Oracle9iAS InterConnect Hub to use:

- Host Name—The hostname of the computer on which Oracle9iAS InterConnect Hub is installed.
- Port Number—The port number of the computer.
- Database SID—The system identifier (SID) of the Oracle9iAS InterConnect Oracle9iAS Metadata Repository.
- Password—The password for the Oracle9iAS Metadata Repository schema.

The Oracle9iAS Metadata Repository stores metadata used by Oracle9iAS InterConnect to coordinate communication between components.

7. Click **Next**. The Oracle9iAS InterConnect Adapter Configuration page displays. Enter the application to be defined or already defined in iStudio in the Application Name field. White spaces or blank spaces are not permitted. The default value is `mySIEBELApp`.

8. Click **Next**. Complete the fields for any other components selected for installation, such as other adapters. When finished, the Summary page displays.

9. Click **Install** to install the Siebel adapter and other selected components. The Siebel adapter is installed in the following directory:

Platform	Directory
Windows	%ORACLE_HOME%\oai\9.0.2\adapters\ <i>Application</i>
UNIX	\$ORACLE_HOME/oai/9.0.2/adapters/ <i>Application</i>

Application is the value you specified in Step 8 on page 2-3.

Siebel Adapter Configuration

Table 2-2, Table 2-3, and Table 2-4 describe executable files, configuration files, and directories. These files and directories are accessible from the directory shown in Table 2-1:

Table 2-1 Siebel Adapter Directory

On...	Go to...
UNIX	\$ORACLE_HOME/oai/9.0.2/adapters/ <i>Application</i>
Windows	%ORACLE_HOME%\oai\9.0.2\adapters\ <i>Application</i>

Table 2-2 Executable Files

File	Description
start.bat (Windows)	Takes no parameters, starts the adapter.
start (UNIX)	
stop.bat (Windows)	Takes no parameters, stops the adapter.
stop (UNIX)	
ignoreErrors.bat (Windows)	If an argument is specified, then the given error code will be ignored. If no argument is specified, then all error codes specified in the <code>ErrorCodes.ini</code> will be ignored.
ignoreErrors (UNIX)	

Table 2–3 Configuration Files

File	Description
ErrorCodes.ini (Windows and UNIX)	Should contain one error code per line.
adapter.ini (Windows and UNIX)	Consists of all the initialization parameters which the adapter reads at startup. Refer to Appendix A for a typical adapter.ini file.

Table 2–4 Directories

File	Description
persistence	The messages are persisted in this directory. This directory or its contents should not be edited.
logs	The logging of adapter activity is done in subdirectories of the log directory. Each new run of the adapter creates a new subdirectory in which logging is done in an oailog.txt file.

Using the Application Parameter

Adapters do not have integration logic. The Siebel adapter has a generic transformation engine that processes metadata from the repository as runtime instructions to do transformations. The application defines for an adapter what its capabilities are. For example, it can define what messages it can publish, what messages it can subscribe to, and what are the transformations to perform. The application parameter allows the adapter to become smart in the context of the application to which it is connected. It allows the adapter to retrieve from the repository only that metadata that is relevant to the application. The application parameter must match the corresponding application that will be defined in *iStudio* under the Applications folder.

If you are using pre-packaged metadata, after importing the pre-packaged metadata into the repository, start up *iStudio* to find the corresponding application (under the Applications folder in *iStudio*) to use as the application for the adapter you are installing (unless the package you are using provides directions for what the application should be).

adapter.ini Initialization Parameter File

This section contains these topics:

- [Hub.ini](#)
- [Agent Connection Parameters](#)
- [Siebel Adapter Parameters](#)

Hub.ini

The Siebel adapter connects to the hub database using parameters from the `hub.ini` file located in the hub directory. The following table lists the parameter name, a description for each parameter, the possible and default values, and an example.

Parameter	Description	Example
<code>hub_username</code>	The name of the hub database schema (or username). Possible values are valid hub database username. There is no default value.	<code>hub_username=myhub</code>
<code>hub_password</code>	The password for the hub database user. Possible values are the valid password for the hub database user. There is no default value.	<code>hub_password=manager</code>
<code>hub_host</code>	The name of the machine hosting the hub database. Possible values are the valid machine name. There is no default value.	<code>hub_host=mpjoshipc</code>
<code>hub_instance</code>	The valid SID of the hub database. There is no default value.	<code>hub_instance=orcl</code>
<code>hub_port</code>	The TNS listener port number for the HUB database instance. There is no default value.	<code>hub_port=1521</code>
<code>repository_name</code>	The valid name of the repository this adapter talks to. There is no default value.	<code>repository_name=myrepo</code>

Agent Connection Parameters

The Siebel adapter connects to the spoke application using parameters from the `adapter.ini` file. The following table lists the parameter name, a description for each parameter, the possible and default values and an example.

Parameter	Description	Example
<code>application</code>	The name of the application this adapter connects to. This must match with the name specified in iStudio during creating of metadata. Any alphanumeric string can be used. There is no default value.	<code>application=aqapp</code>
<code>partition</code>	The partition this adapter handles as specified in iStudio. Any alphanumeric string is a possible value. There is no default value.	<code>partition=germany</code>
<code>instance_number</code>	To have multiple adapter instances for the given application with the given partition, each adapter should have a unique instance number. Possible values are any integer greater than 1. There is no default value.	<code>instance_number=1</code>
<code>agent_log_level</code>	Specifies the amount of logging necessary. Possible values are: 0=errors only 1=status and errors 2=trace, status, and errors The default value is 1.	<code>agent_log_level=2</code>
<code>agent_subscriber_name</code>	The subscriber name used when this adapter registers its subscription. The possible value is a valid Oracle Advanced Queuing subscriber name and there is no default value.	<code>agent_subscriber_name=aqapp</code>
<code>agent_message_selector</code>	Specifies conditions for message selection when registering its subscription with the hub. The possible value is a valid Oracle Advanced Queuing message selector string. There is no default value.	<code>agent_message_selector=recipient_list like '%aqapp,%'</code>
<code>agent_reply_subscriber_name</code>	The subscriber name used when multiple adapter instances for the given application with the given partition are used. Optional if there is only one instance running. The possible value is application name (parameter: <code>application</code>) concatenated with instance number (parameter: <code>instance_number</code>). There is no default value.	If <code>application=aqapp</code> , <code>instance_number=2</code> , then, <code>agent_reply_subscriber_name=aqapp2</code>

Parameter	Description	Example
agent_reply_message_selector	Used only if multiple adapter instances for the given application with the given partition. The possible value is a string built using concatenating application name (parameter:application) with instance number (parameter:instance_number). There is no default value.	If application=aqapp, instance_number=2, then agent_reply_message_selector=recipient_list like '%,aqapp2,%'
agent_tracking_enabled	Specifies if message tracking is enabled. Set to false to turn off all tracking of messages. Set to true to track messages with tracking fields set in iStudio. Possible values are true or false. The default value is true.	agent_tracking_enabled=true
agent_throughput_measurement_enabled	Specifies if throughput measurement is enabled. Set to true to turn on all throughput measurements. Possible values are true or false. The default value is true.	agent_throughput_measurement_enabled=true
agent_use_custom_hub_dtd	Specifies if a custom DTD should be used for the common view message when handing it to the hub. By default adapters use an Oracle9iAS InterConnect-specific DTD for all messages sent to the hub as other Oracle9iAS InterConnect adapters will be retrieving the messages from the hub and know how to interpret them. Set to true if for every message, the DTD imported for the message of the common view is to be used instead of the Oracle9iAS InterConnect DTD. Only set to true if a Oracle9iAS InterConnect adapter is not receiving the messages from the hub. Possible values are true or false. There is no default value.	agent_use_custom_hub_dtd=false
agent_metadata_caching	Specifies the metadata caching algorithm. Possible values are: <ul style="list-style-type: none"> ■ startup—Cache everything at startup. This may take a while if there are a lot of tables in the repository. ■ demand—Cache metadata as it is used. ■ none—No caching. This slows down performance. The default value is demand.	agent_metadata_caching=demand

Parameter	Description	Example
agent_dvm_table_caching	<p>Specifies the DVM caching algorithm. Possible values are:</p> <ul style="list-style-type: none"> ▪ startup—Cache all DVM tables at startup. This may take a while if there are a lot of tables in the repository. ▪ demand—Cache tables as they are used. ▪ none—No caching. This slows down performance. <p>The default value is demand.</p>	agent_dvm_table_caching=demand
agent_lookup_table_caching	<p>Specifies the lookup table caching algorithm. Possible values are:</p> <ul style="list-style-type: none"> ▪ startup—Cache all lookup tables at startup. This may take a while if there are a lot of tables in the repository. ▪ demand—Cache tables as they are used. ▪ none—No caching. This slows down performance. <p>The default value is demand.</p>	agent_lookup_table_caching=demand
agent_delete_file_cache_at_startup	<p>With any of the agent caching methods enabled, metadata from the repository is cached locally on the file system.</p> <p>Set this parameter to <code>true</code> to delete all cached metadata on startup.</p> <p>Note: After changing metadata or DVM tables for this adapter in iStudio, you must delete the cache to guarantee access to the new metadata or table information.</p> <p>Possible values are <code>true</code> or <code>false</code>. The default value is <code>false</code>.</p>	agent_delete_file_cache_at_startup=false
agent_max_ao_cache_size	Specifies the maximum number of application objects' metadata to cache. Possible values are any integer greater than 1. The default value is 200.	agent_max_ao_cache_size=200
agent_max_co_cache_size	Specifies the maximum number of common objects' metadata to cache. Possible values are any integer greater than 1. The default value is 100.	agent_max_co_cache_size=100
agent_max_message_metadata_cache_size	Specifies the maximum number of messages' metadata to cache (publish/subscribe and invoke/implement). Possible values are any integer greater than 1. The default value is 200.	agent_max_message_metadata_cache_size=200

Parameter	Description	Example
agent_max_dvm_table_cache_size	Specifies the maximum number of DVM tables to cache. Possible values are any integer greater than 1. The default value is 200.	agent_max_dvm_table_cache_size=200
agent_max_lookup_table_cache_size	Specifies the maximum number of lookup tables to cache. Possible values are any integer greater than 1. The default value is 200.	agent_max_lookup_table_cache_size=200
agent_max_queue_size	Specifies the maximum size that internal Oracle9iAS InterConnect message queues can grow. Possible values are any integer greater than 1. The default value is 1000.	agent_max_queue_size=1000
agent_persistence_queue_size	Specifies the maximum size that internal Oracle9iAS InterConnect persistence queues can grow. Possible values are any integer greater than 1. The default value is 1000.	agent_persistence_queue_size=1000
agent_persistence_cleanup_interval	Specifies how often the persistence cleaner thread should run. Possible values are any integer greater than 30000. The default value is 60000.	agent_persistence_cleanup_interval=60000
agent_persistence_retry_interval	Specifies how often the persistence thread should retry when it fails to push a Oracle9iAS InterConnect message. Possible values are any integer greater than 5000. The default value is 60000.	agent_persistence_retry_interval=60000
service_path	Windows only. The value that the environment variable PATH should be set to. Path is set to the specified value before forking the Java VM. Typically, all directories containing all necessary DLLs should be listed here. Possible values are the valid path environment variable setting. There is no default value.	service_path=%JREHOME%\bin;D:\oracle\ora902\bin
service_classpath	The classpath used by the adapter Java VM. If a custom adapter is developed and as a result, the adapter is to be used to pick up any additional jars, add the jars to the existing set of jars being picked up. Possible values are the valid classpath. There is no default value.	service_classpath=D:\oracle\ora902\oai\902\lib\oai.jar;%JREHOME%\lib\i18n.jar;D:\oracle\ora902\jdbc\classes12.zip
service_class	The entry class for the Windows NT service. The possible value is oracle/oai/agent/service/AgentService. There is no default value.	service_class=oracle/oai/agent/service/AgentService
service_max_java_stack_size	Windows only. The maximum size to which the Java VM's stack can grow. Possible values are the valid Java VM maximum native stack size. The default value is the default for the Java VM.	service_max_java_stack_size=409600

Parameter	Description	Example
service_max_native_stack_size	Windows only. The maximum size to which the Java VM's native stack can grow. Possible values are the valid Java VM maximum native stack size. The default value is the default for the Java VM.	service_max_native_size=131072
service_min_heap_size	Windows only. Specifies the minimum heap size for the adapter Java VM. Possible values are the valid Java VM heap sizes. The default value is the default Java VM heap size.	service_min_heap_size=536870912
service_max_heap_size	Windows only. Specifies the maximum heap size for the adapter Java VM. Possible values are any valid Java VM heap sizes. The default value is 536870912.	service_max_heap_size=536870912
service_num_vm_args	Windows only. The number of <code>service_vm_arg<number></code> parameters specified. Possible values are the number of <code>service_vm_arg<number></code> parameters. There is no default value.	service_num_vm_args=1
service_vm_arg<number>	Windows only. Specifies any additional arguments to the Java VM. For example, to get line numbers in any of the stack traces, set <code>service_vm_arg1=java.compiler=NONE</code> . If there is a list of arguments to specify, use multiple parameters as shown in the example by incrementing the last digit starting with 1. Be sure to set the <code>service_num_vm_args</code> correctly. Possible values are any valid Java VM arguments. There is no default value.	service_vm_arg1=java.compiler=NONE service_vm_arg2=oai.adapter=.aq
service_jdk_version	Windows only. The JDK version the adapter Java VM should use. The default value is 1.3.1.	service_jdk_version=1.3.1
service_jdk_dll	Windows only. The dll the adapter Java VM should use. The default value is <code>jvm.dll</code> .	service_jdk_dll=jvm.dll

Siebel Adapter Parameters

The following table lists the parameters specific to the Siebel adapter.

Parameter	Description	Example
bridge_class	This indicates the entry class for the Siebel adapter. Do not modify this value. A possible value is <code>com.actional.oai.Agent</code> . There is no default value.	bridge_class=com.actional.oai.Agent

Starting an the Siebel Adapter

Start the Siebel adapter using the `start` script in the directory named after the Siebel adapter on Windows NT, UNIX, or HP.

On Windows NT or Windows 2000, start it from the Service window available from the Start menu.

1. Access the Services window from the Start menu:

On...	Choose...
Windows NT	Start > Settings > Control Panel > Services
Windows 2000	Start > Settings > Control Panel > Administrative Tools > Services

The Services window displays.

2. Select the *OracleHome9iASInterConnectAdapter-Application* service.
3. Start the service based on your operating system:

On...	Choose...
Windows NT	Choose Start.
Windows 2000	Right click the service and choose Start from the menu that displays.

Supported Siebel Interfaces

The Siebel adapter enables Oracle9iAS InterConnect users to use the Siebel eBusiness Application to maintain/synchronize customer, product, pricing, and sales order information.

This chapter discusses the following topics:

- [Siebel Interfaces](#)
- [Creating an Implemented Procedure](#)
- [Creating a Subscribed Event](#)

Siebel Interfaces

The Siebel adapter allows you to browse and call Siebel Business Services (objects with methods) and Business Components (data sets which you can query and update).

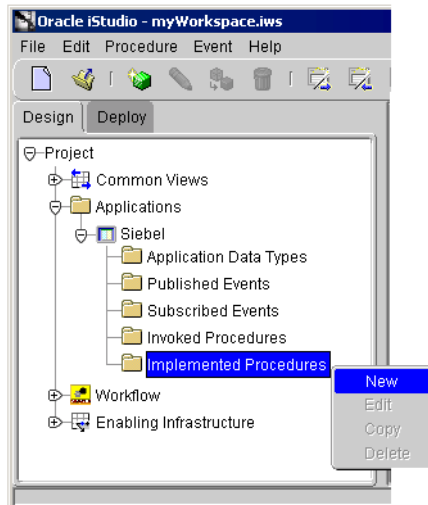
Business Components are representations of a set of one or more joined tables. They have certain “table like” behaviors such as the ability to query a set of records. Business Services are effectively containers for code. Custom business services can be defined using the Siebel Tools, including defining a list of arguments and data types.

Creating an Implemented Procedure

To create an implemented procedure using iStudio:

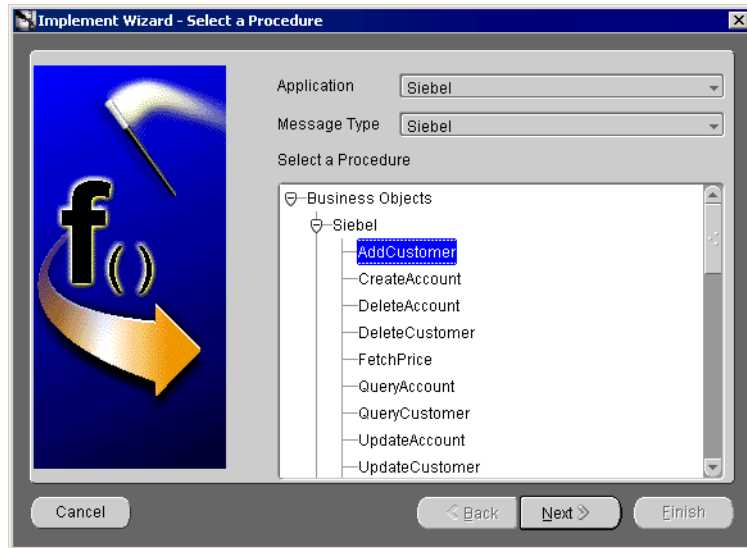
1. Start iStudio.
2. Open your project.
3. Expand the Applications folder.
4. Right-click **Implemented Procedures** and select **New**.

Figure 3–1 *Creating an Implemented Procedure*



The Implement Wizard—Select a Procedure dialog displays.

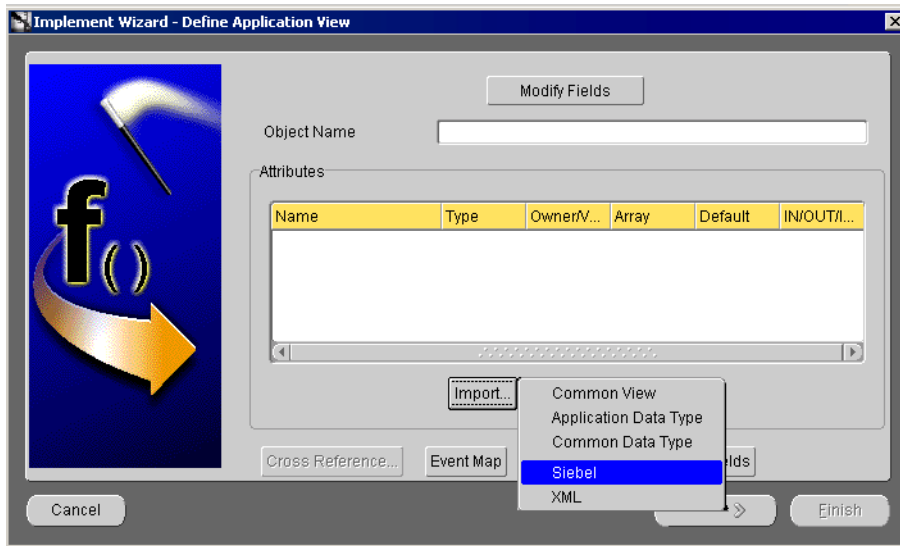
Figure 3–2 *Selecting a Procedure*



5. Select the Application and Message type from the dropdown menus.
6. Select a procedure and click **Next**.

The Implement Wizard—Define Application View dialog displays.

Figure 3–3 *Defining the Application View - Importing Siebel*



7. Import attributes from Siebel. You must log on to Siebel.
If this is the initial login for this machine, enter the correct information.

See Also: ["Importing Attributes from Siebel"](#) on page 3-5

If this machine has been logged in to Siebel before, enter the password on the Siebel Login dialog and click **OK**.

Importing Attributes from Siebel

When you use iStudio to import attributes from Siebel, you must first log on to Siebel. When logging into Siebel from iStudio, the login fields automatically populate, leaving the Password field the only field that requires input.

To import attributes from Siebel:

1. Click **Import** and select **Siebel** on the Define Applications View dialog.

The Siebel Login dialog displays.

The first time you log on from a new workstation, you are required to enter information in every field that is required for your setup. Subsequent logins from that workstation only require a password to log on.

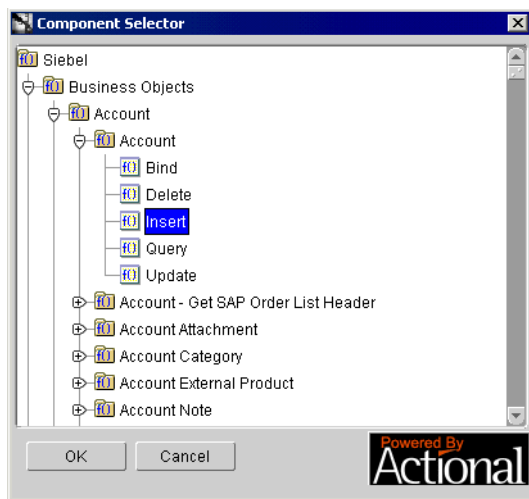
Figure 3–4 Siebel Login Dialog

Enterprise	siebel
Gateway	SDK-LAB2
App Server	SDK-LAB2
Object Manager	SCCObjMgr
User Name	SAdmin
Password	

2. If required, enter information in the following fields:
 - Enterprise—The name of the Siebel Enterprise server.
 - Gateway—The name of the Siebel Gateway server.
 - App Server—The name of the Siebel Application server.
 - Object Manager—The name of the Object Manager file that will be used.
 - User Name—The user name for the system.
 - Password—The user password for the system.
3. Click **OK**.

The Component Selector dialog displays with Business Components.

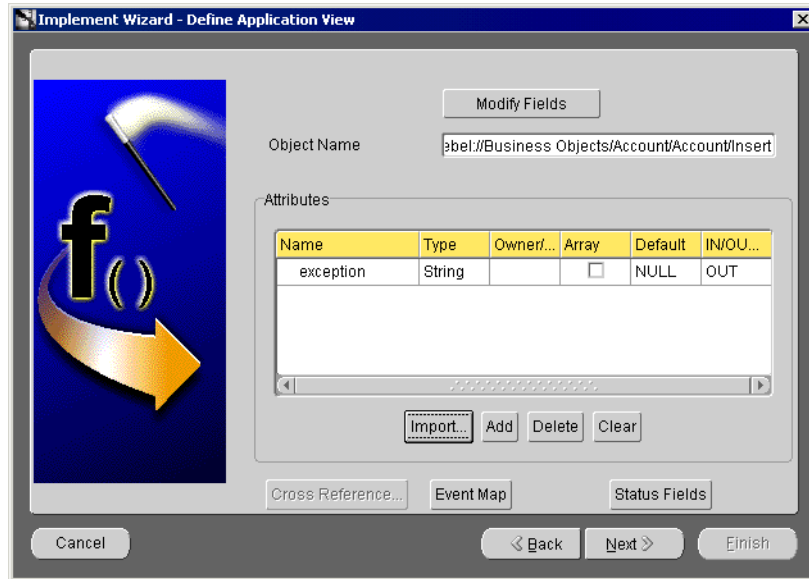
Figure 3-5 Component Selector



4. Expand the Siebel tree until the correct component displays for selection.
5. Select a component and click **OK**.

The Implement Wizard—Define Applications View dialog displays the selected component and its attributes.

Figure 3–6 Populated Define Application View



6. Click Next.

The Define Mappings dialog displays.

7. Click New to define mappings and click Finish.

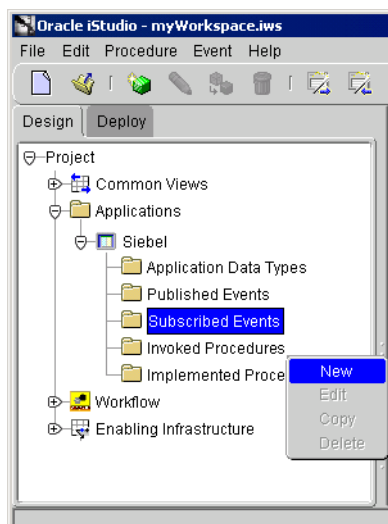
The new populated event displays in the right panel of iStudio.

Creating a Subscribed Event

To create an subscribed event using iStudio:

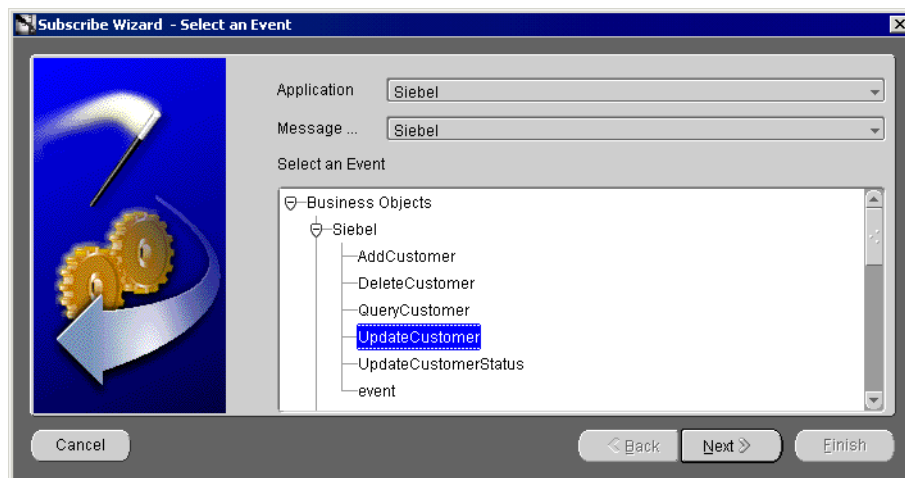
1. Start iStudio.
2. Open your project.
3. Expand the Applications folder.
4. Right-click **Subscribed Events** and select **New**.

Figure 3–7 *Creating a Subscribed Event*



The Subscribe Wizard—Select an Event dialog displays.

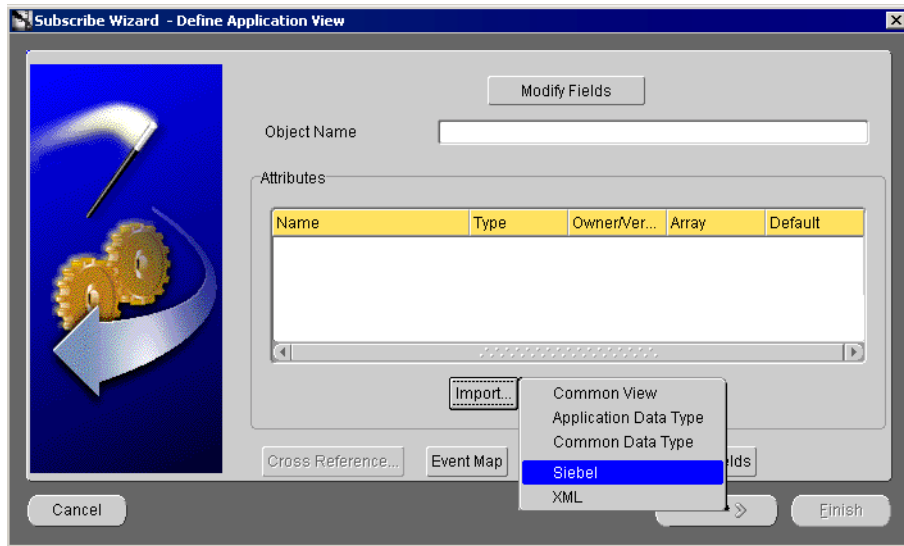
Figure 3–8 *Selecting an Event*



5. Select the Application and Message Type from the dropdown menus.
6. Select an event and click **Next**.

The Define Application View dialog displays.

Figure 3–9 *Defining the Application View - Importing Siebel*



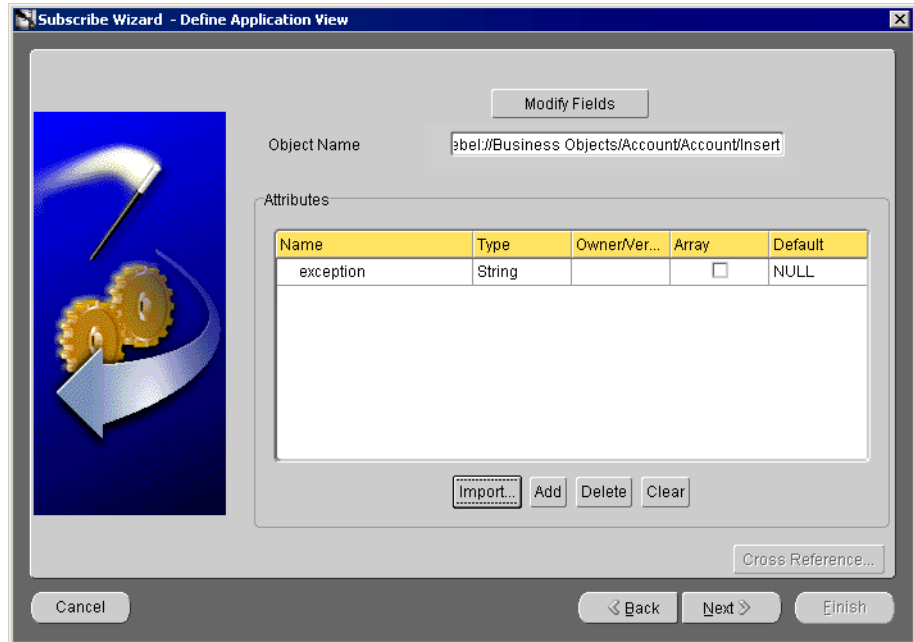
7. Import attributes from Siebel. You must log on to Siebel.
If this is the initial login for this machine, enter the correct information.

See Also: ["Importing Attributes from Siebel"](#) on page 3-5

If this machine has been logged on to Siebel before, enter the password on the Siebel Login dialog and click **OK**.

After attributes are imported, the Define Applications View dialog displays the selected component and its attributes.

Figure 3–10 Populated Subscribe Wizard - Define Application View



8. Click **Next**.

The Define Mappings dialog displays.

9. Click **New** to define the mappings and click **Finish**.

The new populated event displays in the right panel of iStudio.

Business Component Relationships

Under the hood, all Siebel applications use Business Components and Business Services, which are the basic building blocks for Siebel. Custom *methods* (function calls) can be defined using the Siebel Tools, including defining a list of arguments and data types.

This chapter discusses the following topics:

- [Most Commonly Used Siebel Components](#)
- [Component Relationships](#)
- [Handling Multi-Valued Fields](#)
- [Frequently Asked Questions](#)

Most Commonly Used Siebel Components

Various components are accessed by other components across the Siebel solution. Most of the major components interact with other components to provide a logical view of information. The following lists most of the licensed components which are available with any Siebel solution:

- Account
- Business Address
- Contact
- Price List
- Price List Item
- Quote
- Quote Item
- Opportunity
- Opportunity Orders
- Opportunity Product
- Order Entry Orders
- Order Entry Line Items
- RMA
- Service Request

Component Relationships

Siebel has approximately 1650 business components that correspond to various different projects. These, in turn, comprise the Siebel Application and its many modules. Each major component interacts with one or many component(s). Each component that it interacts with can be classified into parent-child relationship. The parent-child relationship among the components is usually a one-to-many relationship. For example:

The Campaign component has a one-to-many relationship with the Account component. The Account component has a one-to-many child relationship with Business Address.

Account

The following is list of components in the Account project.

- Account Attachment—Fields that can be attached to an account.
- Account—The name of the account.
- Account Note—A note that can be attached to an account.
- Account Private Note—A note that can be attached to an account that can only be viewed by specified individuals.
- Account Synonym—An alias attached to an account.

The following is a list of components in the Account project that are component-specific:

- Back Office Account
- Baseline
- Fulfillment Shipper
- Organization Unit
- Plan Account
- Store Conditions
- Sub Account

All the components listed in the Account project have an Account ID field and an Account ID value that is used as the primary ID. An Account ID can be used to retrieve related information.

[Table 4-1](#) lists components with which Account interacts.

Table 4-1 Other Components with which Account Interacts

Component	How Account ID is used
Account Category	Account ID is one of the component fields.
Account Synonym	Account ID is one of the component fields.
Business Address	Account ID value is used for Business Address ID.
Contact	Account ID is one of the component fields.
Industry	Account Pulls information based on language code for display purposes.

Table 4–1 Other Components with which Account Interacts

Component	How Account ID is used
Organization Unit Type	Account ID value is used as primary ID.
Position	Account Name is one of the component fields.
Service Agreement	Account ID is one of the component fields.
Territory	Account Name value is used for Named Account.

Account Parent Components

The following lists important components that have a one-to-many relationship with Account and are component-specific:

- Account
- Action
- Asset Mgmt Asset
- Calendar
- Campaign
- Channel partner
- Comm Request
- Contact
- Contact Category
- Expense Item
- Position
- Service Agreement
- Service Request

Account Child Components

The following lists components with which Account has a one-to-many relationship and are component-specific:

- Account
- Account Get SAP order list output
- Account Attachment

- Account Category
- Account Note
- Account Private Note
- Action
- Activity Plan
- Agreement
- Asset Mgmt-Asset
- Business Address
- Contact
- Customer Product
- Customer Survey
- Industry
- Opportunity
- Order Entry Orders
- Organization Unit Type
- Position
- Quote
- Sales Assessment
- Service Activity
- Service Request
- Sub Account

Business Address

The Account ID field value is used for the ID field in Business Address. The Account object needs to be created before the Address record entry is created. Business Address has no child relationships. The only relationship it has to others is that of parent.

Business Address Parent Components

Listed below are some important components that have a one-to-many relationship with Business Address and are component-specific:

- Account
- Asset Mgmt Asset
- RMA
- Service Agreement
- Service Request

Contact

The following is a list of components in the Contact project that are component-specific:

- Business Address
- Contact Attachment
- Contact Note
- Contact Private Note
- Contact Relationship
- Contact User
- Contact-Prospect Campaigns
- Personal Address
- Personal Payment Profile
- TAS Contact Behavior

All the components other than Business Address, Personal Address, Personal Payment Profile, and Contact User have the Contact ID as one of the fields, which can be used to retrieve related information. Contact Relationship maintains the relationship table between Account and Contact. It has both Contact and Account ID fields.

[Table 4-2](#) lists the field mappings for Business Address.

Table 4-2 Business Address Field Mapping

Business Address Components	Mapping
Address Name	Primary Address Name in Contact.
City	Primary City in Contact.
State	Primary State in Contact.
Postal Code	Primary Postal code in Contact.

[Table 4-3](#) lists other major components with which Contact interacts.

Table 4-3 Contact Components and Interactions

Component	Interaction
Position	Active First Name maps to Employee First Name in Contact. Active Last Name maps to Employee Last Name in Contact. Active Login Name maps to Sales Rep in Contact. Contact Row Status maps to Row Status in Contact. Name maps to Position in Contact.
Contact Category	Category maps to Category Value in Contacts. Category Value maps to Category Value in Contacts.
Opportunity	Opportunity Id is one of the component fields. Name maps to Opportunity in Contact.
Personal Address	Street Address maps to Personal Street Address in Contact. City maps to Personal City in Contact. Country maps to Personal Country in Contact. Postal Code maps to Personal postal Code in Contact. State maps to Personal State in Contact.

Table 4–3 Contact Components and Interactions

Component	Interaction
Account	Relationship is through Contact Relationship.
Employee	Employee Login Name maps to Login in Employee.

Contact Parent Components

The following lists some important components that have a one-to-many relationship with Contact that are component-specific:

- Account
- Action
- Asset Mgmt Asset
- Calendar
- Campaign
- Contact
- Contact Category
- Contact Category value
- Correspondence
- Customer Survey
- Expense Item
- Fs Invoice
- Opportunity
- Position
- Project
- Service Agreement
- Service Request

Contact Child Components

The following lists some important components that Contact has a one-to-many relationship with and are component-specific:

- Action
- Activity Plan
- Asset Mgmt Asset
- Contact
- Contact Attachment
- Contact Category
- Contact Note
- Contact Prospect Campaigns
- Customer Surveys
- Opportunity
- Personal Address
- Personal Payment Profile
- Response
- Sales Assessment
- Service Agreement
- Service Request

Internal Product

Internal Product Component has the following ID fields which can be used to query related information:

- Product ID
- Part #
- Vendor ID
- Vendor Part Number
- Volume Discount ID

[Table 4-4](#) lists the field mappings of Internal Product and other components with which it interacts:

Table 4-4 Internal Products and Other Components with which it Interacts

Internal Products/Components	Maps to
Admin Product Line	Name maps to Product Line in Internal Product. Position maps to Position in Internal Product. Position ID to Position ID in Internal Product. ID maps to Product Line ID in Internal Product.
Employee	Login Name maps Employee in Internal Product. Product Relation Type maps to Relation type in Internal Product. Product Skill Level maps to Expertise Level in Internal Product.
Equivalent Product	Name maps to Equivalent Product in Internal Product.
Internal Division	Back Office Distribution Channel maps Back Office Distribution Channel in Internal Product. Back Office Sales Organization maps Back Office Sales Organization in Internal Product. Min Order Units maps Back Office Min Order Units in Internal Product. Name maps Organization in Internal Product. Organization ID maps Organization ID in Internal Product. Unit of Measure maps Back Office Unit of Measure in Internal Product.

Price List

The following is a list of components in the Price List project that are component-specific:

- Price List Item—Has the Price List ID.
- Volume Discount—The Component Volume Discount field Discount Method maps to the Volume Discount method in Price List, as Name maps to Volume Discount in Price List.
- Volume Discount Item—Has Volume Discount ID which maps to Volume Discount.

[Table 4-5](#) lists major components with which Price List interacts.

Table 4-5 Components with which Price List Interacts

Components	Description
Employee	Login maps to Login name in Price List.
Internal Product	ID maps to Price List Item Product ID in Price List.
Internal Division	Organization ID maps to Organization ID in Price List.
Payment Terms	Name maps to Payment term in Price List.

Price List Parent Components

An important component that has a one-to-many relationship with Price List is Internal Product. Internal Product is component-specific.

Price List Child Components

The following lists some important components that Price List may have a one-to-many relationship and are component-specific:

- Internal Division
- Internal Product
- The relation

Price List Item

Price list Item contains Product ID and Price List ID fields, which can be used to associate it with a Price list or Product. [Table 4-6](#) lists the components and their interactions:

Table 4-6 Components with which Price List Interacts

Component	Interacts with
Price List	Price List ID field does exist in Price List Item.
Volume Discount Item	Volume Discount ID Method maps to Volume Discount ID in Price List Item. Volume Discount Method maps to Volume Discount Method in Price List Item. Volume Discount End Date maps to Volume Discount Method in Price List Item.

Price list Item component has no parent or child relationships with other components.

Quote

The following is a list of components, other than Quote, in the Quote project that are component-specific:

- Quote Attachment
- Quote Item
- Quote Item.Line Number (Sequence)
- Quote Solution
- Quote.Revision (Sequence)

All the components in the project have a Quote ID field, which can be used to associate them and to query associated values.

[Table 4-7](#) lists quote components with which Quote interacts.

Table 4-7 Other components with which Quote Interacts

Quote Components	Description
Business Address	City field maps to Bill to City field in Quote. Country field maps to Bill to Country field in Quote. Postal Code maps to Bill to Postal Code field in Quote. State maps to Bill to State field in Quote. Street Address maps to Bill to Street Address field in Quote.
Contact	ID maps to Contact ID field in Quote. First Name maps to Ship To First Name field in Quote. Last Name maps to Ship To Last Name field in Quote. First name maps to Bill To First Name field in Quote. Last Name maps to Bill To Last Name field in Quote.
Account	Account ID maps to Account ID in Quote.

Table 4–7 Other components with which Quote Interacts

Quote Components	Description
Opportunity Product	Product maps to Opportunity Product field in Quote. Product ID maps to Opportunity Product ID field in Quote. Quote Number maps to Opportunity Quote Number in Quote.

Quote Parent Components

The following lists some important components that have a one-to-many relationship with Quote and are component-specific:

- Account
- Comm Request
- Opportunity
- Position

Quote Child Components

The following lists some important component-specific components with which Quote has a one-to-many relationship:

- Cfg Price list item
- Comm Contact
- Comm Employee
- Opportunity Product
- Line Number
- Order Entry-Orders
- Position
- Service Agreement
- Quote Attachment
- Quote Item
- Quote Item.Number (Sequence)
- Quote Solution
- Quote.Revision (Sequence)

Quote Item

Quote Item defines a line item in a Quote. For example, a quote item needs to be defined for each product in a quote. It contains many ID fields such as Price List ID, Quote ID, Promotion ID, Product Integration ID, Solution ID, Solution Product ID, and Class Reg Num of which can be used to associate and query. It interacts with the following components:

- Product Line—Name maps to the Product Line field in Quote Item.
- Quote Solution—Quote ID maps to Quote ID in Quote Item.
- Volume Discount Item—Volume Discount ID maps to the Volume Discount ID field in Quote Item.
- Product Line—Name maps to the Product Line field in Quote Item.

Quote Item Parent Components

The following lists component-specific components that have a one-to-many relationship with Quote item:

- Quote
- Quote Solution

Quote Item component does not have child relationships with other components.

Opportunity

The following is a list of component-specific components in the Opportunity project other than Opportunity:

- Opportunity Orders
- Opportunity Attachment
- Opportunity Note
- Opportunity Position
- Opportunity Private Note
- Opportunity Product
- Opportunity Product.Line Number (Sequence)
- Opportunity Skill
- Opportunity Skill Item

- Opportunity Skill.Name (Sequence)
- TAS Account Opportunity
- TAS Opportunity Assessment
- TAS Opportunity Competitor

The Opportunity ID field is in all Opportunity project Components other than Opportunity Orders and TAS Opportunity Competitor.

In case of Opportunity Orders, the common field is Parent Opportunity ID field that is common.

TAS Account Opportunity has Account ID, Organization ID, and Position ID, which exists in Opportunity. These can be used to obtain any cross-reference data that is needed.

[Table 4–8](#) lists other components with which TAS Account Opportunity interacts.

Table 4–8 Other components with which TAS Account Opportunity Interacts

Component	Description
Assignment Group	Name maps to Territory field in Opportunity.
Business Address	City maps to City Field in Opportunity. CityStateZipCountry maps to CityStateZipCountry Field in Opportunity. Country field maps to Country field in Opportunity. Postal Code maps to Postal Code field in Opportunity. State maps to State field in Opportunity. Street Address maps to Street Address field in Opportunity.
Competitor	Vendor maps to Competitor field in Opportunity.
Industry	Name maps to Industry field in Opportunity.
Position	Active First Name maps to First Name field in Opportunity. Active Last Name maps to Last Name field in Opportunity. Sales Rep maps to Login Name field in Opportunity. Name maps to Position field in Opportunity. ID maps to Position ID field in Opportunity.

Table 4–8 Other components with which TAS Account Opportunity Interacts

Component	Description
Source	End Date maps to Program End Date field in Opportunity. Start Date maps to Program Start Date field in Opportunity. Name maps to Source field in Opportunity. Created maps to Source Created Date field in Opportunity. Type maps to Source Type field in Opportunity.

Opportunity Parent Components

The following lists some important component-specific components that have a one-to-many relationship with Opportunity:

- Account
- Channel Partner
- Comm Request
- Consumer
- Contact
- Opportunity
- Opportunity Category
- Position
- Recipients

Opportunity Child Components

The following lists some important component-specific components that Opportunity maintains a one-to-many relationship:

- Account
- Activity Plan
- Campaign
- Campaign Contact
- Competitor
- Contact

- Opportunity
- Opportunity Attachment
- Opportunity Forecast
- Opportunity Note
- Opportunity Product
- Opportunity Skill
- Proposal
- Quote
- Sales Assessment

Opportunity Orders

Opportunity Orders has the following ID fields that can be used to reference, query, and associated values:

- Account ID
- Organization ID
- Parent Opportunity ID
- Position ID
- Sales Method ID
- Primary Territory ID
- Primary Competitor ID

[Table 4-9](#) lists other components with which Opportunity Orders interacts.

Table 4–9 Other components with which Opportunity Orders Interacts

Opportunity Orders Components	Components
Position	Active First Name maps to First Name field in Opportunity Orders. Active Last Name maps to Last Name field in Opportunity Orders. Sales Rep maps to Login Name field in Opportunity Orders. Name maps to Position field in Opportunity Orders. ID maps to Position ID field in Opportunity Orders.
Business Address	City maps to City Field in Opportunity Orders. CityStateZipCountry maps to CityStateZipCountry Field in Opportunity Orders. Country field maps to Country field in Opportunity Orders. Postal Code maps to Postal Code field in Opportunity Orders. State maps to State field in Opportunity Orders. Street Address maps to Street Address field in Opportunity Orders.
Competitor	Vendor maps to Competitor field in Opportunity Order.
Opportunity Product	Product maps to Product field in Opportunity Order.
Source	End Date maps to Program End Date field in Opportunity Order. Start Date maps to Program Start Date field in Opportunity Order. Name maps to Source field in Opportunity Order. Created maps to Source Created Date field in Opportunity Order. Type maps to Source Type field in Opportunity Order.
Territory	Name maps to Territory field in Opportunity Order.

Opportunity Orders does not have parent or child relationships with other components.

Opportunity Product

Opportunity product has the following components through which all the data related to the product and data associated with other components can be retrieved:

- Product ID
- Oppty ID
- Part Number
- Territory ID
- Vendor ID
- Quote Number

[Table 4–10](#) lists other components with which Opportunity Product interacts.

Table 4–10 Other Components with which Opportunity Product Interacts

Opportunity Product Components	Components
Position	<p>Sales Rep maps to Login Name field in Opportunity Product.</p> <p>Closed Date maps to Opportunity Closed Date field in Opportunity Product.</p> <p>Revenue maps to Opportunity Revenue field in Opportunity Product.</p> <p>Revenue Currency Code maps to Opportunity Revenue Currency Code field in Opportunity Product.</p> <p>Revenue Exchange Date maps to Opportunity Revenue Exchange Date field in Opportunity Product.</p>
Territory	<p>Name maps to Territory field in Opportunity Product.</p>

Opportunity Product Parent Components

The following lists some component-specific components that have a one-to-many relationship with Opportunity Product:

- Internal Product
- Opportunity

Opportunity Product Child Components

The following lists some important component-specific components that Opportunity Product maintains a one-to-many relationship:

- Position
- Territory

Order Entry Orders

The Orders component is an important component in Orders project. The other component-specific components in the Orders other than Order Entry include:

- Order Entry—Line Item Action Types
- Order Entry—Line Item Actions
- Order Entry—Line Item Available Products
- Order Entry—Line Item Available Substitutes
- Order Entry—Line Items
- Order Entry—Line Item Line Items.Line Number (Sequence)
- Order Entry—Order Sales Team
- Order Entry—Order Types
- Order Entry—Order Types to Action Types
- Order Entry Attachment

All the project components have Order ID, Order Header ID, or Order Number through which other components can co-relate and relate data.

Other Components With Which Order Entry Interacts

[Table 4-11](#) lists other components with which Orders interacts.

Table 4-11 Components with which Orders Interacts

Component	Description
Account	Account ID maps to Account ID field in Order Entry Orders. Name maps to Account field in Order Entry Orders.

Table 4–11 Components with which Orders Interacts

Component	Description
Business Address	<p>City maps to Bill To—City Field in Order Entry Orders.</p> <p>Country field maps to Bill To—Country field in Order Entry Orders.</p> <p>Postal Code maps to Bill To—Zip field in Order Entry Orders.</p> <p>State maps to Bill To—State field in Order Entry Orders.</p> <p>Street Address maps to Bill To—Address field in Order Entry Orders.</p> <p>City maps to Ship To—City Field in Order Entry Orders.</p> <p>Country field maps to Ship To—Country field in Order Entry Orders.</p> <p>Postal Code maps to Ship To—Zip field in Order Entry Orders.</p> <p>State maps to Ship To - State field in Order Entry Orders.</p> <p>Street Address maps to Ship To—Address field in Order Entry Orders.</p> <p>Country Code maps to Account Address—Country field in Order Entry Orders.</p> <p>State maps to Account Address.</p> <p>State field in Order Entry Orders.</p> <p>Street Address maps to Account Address field in Order Entry Orders.</p>
Contact	<p>First Name maps to Bill To First Name field in Order Entry Orders.</p> <p>Last Name maps to Bill To Last Name field in Order Entry Orders.</p> <p>First Name maps to Ship To First Name field in Order Entry Orders.</p> <p>Last Name maps to Ship To Last Name field in Order Entry Orders.</p>
Invoice Payments	<p>Payment Import Flg maps to Payment Import Flg field in Order Entry Orders.</p>
Employee	<p>Login maps to Approved By Emp Login field in Order Entry Orders.</p> <p>ID maps to Approved By—Employee ID field in Order Entry Orders.</p>
Order Entry - Line Items	<p>Order Header ID maps to Order ID field in Order Entry Orders.</p>

Order Entry Orders Parent Components

The following lists some important component-specific components that have a one-to-many relationship with Order Entry Orders:

- Account
- Comm Request
- Quote
- Service Request

Order Entry Orders Child Components

[Table 4-12](#) lists some important components with which Order Entry Orders maintains a one-to-many relationship.

Table 4-12 Components with which Order Entry Orders Maintains a One-to-Many Relationship

Component	Description
Comm Contact	Component-specific
FS Invoice Payments	Component-specific
Order Entry—Create SAP Order Output	Component-specific
Order Entry—Create SAP Order Status Header	Component-specific
Order Entry—Get SAP Order Status Item Delivery Item	Component-specific
Order Entry—Get SAP Order Status Output	Component-specific
Order Entry—Line Item	Component-specific
Order Entry—Line Items.Line Number (Sequence)	Component-specific
Order Entry—Order Sales Team	Component-specific
Order Entry AttachmentPosition	Component-specific

Order Entry Line Items

The Order Entry Line Item component describes a line item in an Order. It has various Ids, such as Order Header ID, Order Acct ID, Contact ID, Asset ID, Covered Product ID, Volume Discount ID, and Price List ID through which other components can be accessed and obtain and update related data.

[Table 4-13](#) lists other components with which Order Entry Line Items interacts.

Table 4-13 Other components with which Order Entry Line Items Interacts

Component	Description
Price List	ID maps to Price List ID field in Order Entry line Item.
Product Line	Name maps to Product Line field in Order Entry line Item.
Business Address	City maps to Ship To—City Field in Order Entry line Item. Country field maps to Ship To—Country field in Order Entry line Item. Postal Code maps to Ship To—Zip field in Order Entry line Item. State maps to Ship To—State field in Order Entry line Item. Street Address maps to Ship To—Address field in Order Entry line Item.
Contact	First Name maps to Ship To First Name field in Order Entry line Item. Last Name maps to Ship To Last Name field in Order Entry line Item.

Order Entry Line Items Parent Components

The following lists some important component-specific components that have a one-to-many relationship with Order Entry Line Items:

- Campaign Orders (Chart)
- FS Allocated Information
- FS Invoice Line Item Details
- FS Receive RMA PO

- Order Entry Sales Order
- Order Entry Orders
- Response Order Entry Orders

Order Entry Line Items Child Components

The following lists some important component-specific components that have a one-to-many relationship with Order Entry Line Items:

- FS Order Entry Line Item Warranty
- FS Receiving Action RMA PO
- FS Repair
- Order Entry Line Item Actions
- Order Entry Line Item Available Products
- Order Entry Line Item Available Substitutes

RMA

The following lists component-specific components in the RMA project other than RMA:

- RMA Affected Product
- RMA Loaned Product
- RMA Replacement Product

All the components in the project have RMA ID, which can be used to query and associate related data.

[Table 4-14](#) lists other components with which RMA interacts.

Table 4-14 Other components with which RMA Interacts

Component	Description
Business Address	City maps to City Field in RMA. Country field maps to Country field in RMA. Postal Code maps to Postal Code field in RMA. State maps to State field in RMA. Street Address maps to Address field in RMA.

Table 4–14 Other components with which RMA Interacts

Component	Description
Account	ID maps to Account ID field in RMA. Name Code maps to Account field in RMA.
Contact	ID maps to Contact ID field in RMA. First Name maps to Contact First Name field in RMA. Last Name maps to Contact Last Name field in RMA.

RMA Parent Components

The Service Request component has a one-to-many relationship with RMA.

RMA Child Components

The following lists some important components that RMA maintains a one-to-many relationship:

- Business Address
- RMA Affected Product
- RMA Loaned Product
- RMA Replacement Product

Service Request

The following lists component-specific components in the Service project other than Service Request:

- Customer Survey
- SR External Product
- Service Request (Open)
- Service Request Area

All components have SR ID or SREP SR ID or SR Number which can be used to query and insert associated information.

[Table 4–15](#) lists other components with which RMA interacts.

Table 4–15 Components with which RMA Interacts

Component	Description
Action	Owned by maps to Activity Assigned To field in Service Request. Status maps to Activity Status field in Service Request.
Business Address	ID maps to Address ID Field in Service Request. City maps to City Field in Service Request. Country field maps to Country field in Service Request. Postal Code maps to Postal Code field in Service Request. State maps to State field in Service Request. Street Address maps to Address field in Service Request.
Service Agreement	Agreement End Date maps to Agreement End Date Field in Service Request. Agreement Start Date maps to Agreement Start Date Field in Service Request. Agreement Status maps to Agreement Status Field in Service Request. Name maps to Agreement Name Field in Service Request. Revision Number maps to Revision number Field in Service Request. Service Type maps to Service Type Field in Service Request.
SR External Product	Comment maps to Profile Comment Field in Service Request. Description maps to Profile Description Field in Service Request. Name maps to Profile Product Name Field in Service Request. SREP Account ID maps to Profile SREP Account ID Field in Service Request. SREP Product ID maps to Profile SREP Product ID Field in Service Request. Version maps to Profile Version Field in Service Request.
Contacts	Read Flag maps to SR Read Flag Field in Service Request. Red Flag maps to SR Red Flag Field in Service Request.
Account Synonym	Name maps to Synonym field in Service Request.

Service Request Parent Components

The following lists some important component-specific components that have a one-to-many relationship with Service Request:

- Account
- Action
- Admin Resolution Item
- Asset Mgmt Asset
- Comm Request
- Consumer
- Contact
- Correspondence Recipient
- Consumer Product
- Product Defect
- Service request Area
- Solution
- SR Resolution Item

Service Request Child Components

The following lists some important component-specific components that Service Request maintains a one-to-many relationship:

- Account External Product
- Account Synonym
- Action
- Activity Plan
- Asset mgmt - Asset
- Business Address
- Comm Contact
- Comm Employee
- Contact

- Customer Product
- Customer Survey
- FS Invoice
- FS Service Request Symptom
- Order Entry - Orders
- Product Defect
- Resolution Activity
- Service Agreement
- Service Agreement Contact
- Service Request Attachment
- Shift Hour
- Solution SR/PD
- SR External product
- SR Resolution Item

Handling Multi-Valued Fields

A Multi-Valued Field (MVF) is a field that is declared in a primary component but is also available in other Components. The concept is similar to a foreign key in database tables. In the case of Siebel, if you need to update a multi-value field in any component, you need to get the primary component in order to update that value. The Siebel Adapter exposes the Multi Valued Field only in the primary component and not in any other components. This makes the update easier and eases the complexity of updating the MVF Values.

For example, the Account component has the multi valued field `Street Address` whose primary component is `Business Address`. The following sample code adds a new address to the *Hong Kong Flower Shop* account record.

In Siebel this is how you update multi valued fields. The following example is displayed in Siebel VB script:

```
Dim AcctBO as BusObject
Dim AcctBC as BusComp
Dim AddrBC as BusComp
Set AcctBO = theApplication.GetBusObject("Account")
```

```

Set AccntBC = AccntBO.GetBusComp("Account")
With AccntBC
    .SetViewMode SalesRepView
    .ActivateField "Name"
    .ClearToQuery
    .SetSearchSpec "Name", "Hong Kong Flower Shop"
    .ExecuteQuery
Set AddrBC = .GetMVGBusComp("Street Address")
End With
With AddrBC
    .NewRecord NewAfter
    .SetFieldValue "Street Address", "100 main st."
    .SetFieldValue "City", "Denver"
    .WriteRecord
End With
Set AccntBO = Nothing
Set AccntBC = Nothing
Set AddrBC = Nothing

```

Using Siebel, Street Address only appears as a field in Business Address component. You may need to call insert or update for the Business Address component.

```

Dim accRecord(0)           As Business_Objects_AccountRecord
Dim queryAccRec()         As Business_Objects_AccountRecord
Dim accBusAddrRec(0)     As Business_Objects_Business_AddressRecord
Dim queryAccBusAdr()     As Business_Objects_Business_AddressRecord
Dim toDay                 As Variant
Dim expAcct               As ExptdAcctVal
Dim expBusAddr            As ExptdBusAddrsVal
Dim queryExpr             As String

Set accRecord(0) = New Business_Objects_AccountRecord
Set accBusAddrRec(0) = New Business_Objects_Business_AddressRecord

'Set the query expression based on fields Account Name
'lets assume the account name is ACBFirst
'in case we know the Account ID we directly use the
'Business Address Component to call update or insert value

queryExpr = "[Name] = " + Chr$(34) + " Hong Kong Flower Shop" + Chr$(34)

'Queries the insert record based on Value

accountComponent.Query queryExpr, "", queryAccRec

```

```
'Sets values to the fields of
'Business_AddressRecord in case record needs to be inserted
'else just set the changed value in case of update

accBusAddrRec(0).theAccount_Id = queryAccRec (0).theId
accBusAddrRec(0).theStreet_Address = "100 main st."
accBusAddrRec(0).theCity = " Denver "

' Inserts a Business Address Record.
acctBusAddress.Insert accBusAddrRec
accRecord(0)      = nothing
accBusAddrRec(0) = nothing
queryAccRec()    = nothing
queryAccBusAdr() = nothing
```

Frequently Asked Questions

Q) For an account, what is the exact procedure to fetch an account, and then fetch all of the addresses for that account or some other associated child component?

A) Query on Account Component based on the interested Account details. Account and Business Address component are related through ID field. Query on Business Address based on the Account ID value. This should return data about all Business Addresses associated with that Account.

Q) Does a relation exist between an account and its Orders?

A) Account ID is one of the component fields in Order Entry Orders. You can retrieve all Order details with regard to the Account using the Account ID Field.

Q) Does a relation exist between Orders and Prices?

A) The relationship is through the ID field, Order entry has Price List ID field. Order Entry - Orders also has other IDs such as Organization ID, Quote ID, Promotion ID, and Ship To Contact ID. through which you can find other associated details by querying on particular IDs or group of IDs on Order Entry Orders component.

Q) How do I create an account with multiple addresses?

A) Business Address is a child component to Account. Relationship between Account and Business Address is of type, one-to-many. Business Address and Account are related through ID field. Multiple Business Address records with the same Account ID value will be associated with a single Account.

Q) How do I create an order with multiple Line items?

A) Order Entry Line Item component is a child component to Order Entry Orders. Relationship between Order Entry Orders and Order Entry Line Item is of type, one-to-many. Order Entry Line Item and Order Entry Orders are related through Order Header ID field, which is the Order ID. Multiple Order Entry Line Item records with the same Order Header ID value will be associated with a single Order Entry Order. There are also IDs common, such as Order Acct ID, and Contact ID. that can be used to get associated data.

This chapter describes how to use the Configuration Editor to configure the Siebel adapter. The Configuration Editor is only used at runtime.

The following topics are discussed:

- [Configuration Editor](#)
- [Log On Siebel](#)
- [General](#)

Note: Profiles and Deployment are sensitive to the Master Key setting. If using a shared machine, before accessing the Configuration Editor, ensure the Master Key either is set to that of User1 or create a new Master Key for your profiles. Refer to the *Oracle9iAS InterConnect Configuration Editor User's Guide* for more information on the Master Key.

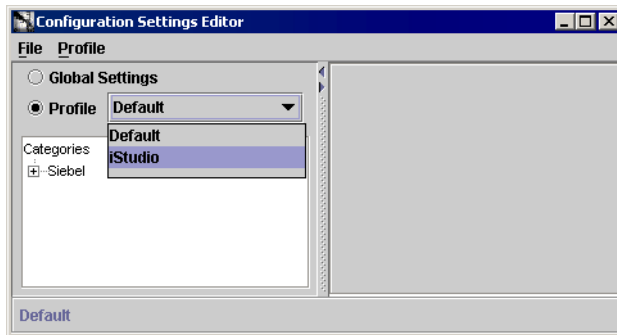
Configuration Editor

To configure settings for the Siebel adapter you must access the Siebel Configuration Editor as follows:

1. Change directories to the `... \oai\9.0.2\config\configeditor` using a DOS prompt.
2. Type **configeditor** and press **Enter**.
The Configuration Editor displays.
3. Click **Profile**.
4. Select the **iStudio** profile.

Note: Under some circumstances you may wish to run your adapter under a profile other than iStudio. This may be needed, for example, if you want to run two instances of the Siebel adapter on the same machine. You may want to have two instances of the same type of adapter if these instances need to connect to different backend system installations. To accomplish this you need to create a new profile using the configuration editor and fill in the settings for this new profile. The name of the new profile should be the same as the name of the application. For example if your application is called APP2, create a profile called APP2. Now APP2 will use the settings in the profile called APP2 whenever it runs.

Figure 5–1 Configuration Settings Editor Default Dropdown Menu



5. Expand **Siebel** to edit the configuration settings for the iStudio profile.

Log On Siebel

The Default Login to Siebel allows you to program your application to automatically connect to Siebel servers. The login credentials under both Repository credentials and Default Runtime Credentials branches are used by the Siebel adapter. The default credentials are used at runtime to submit requests to the business functions.

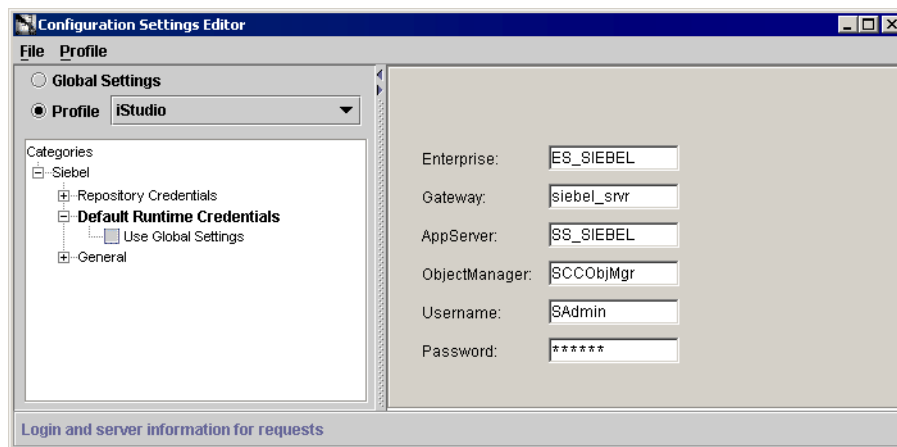
Note: Follow the same procedure to modify the repository credentials.

The Default Login to Siebel authenticates your runtime credentials. From the Configuration Editor main menu:

1. In the left panel, expand **Default Runtime Credentials**.
2. Click **Use Global Settings** to edit the configuration settings for your profile.
3. Click **Default Runtime Credentials**.

The Configuration Settings Editor login dialog displays in the right panel.

Figure 5–2 Configuration Settings Editor Login Dialog



4. Enter the information in the following fields:
 - Enterprise—The Siebel system Enterprise Server.

- Gateway—The name of the gateway. This can be an IP address or a gateway name.
- AppServer—The name of the Application server. This is where there is granted access to all the metadata for the Siebel system. The metadata is the information regarding the Business Services, Business Objects, and the Business Components.
- ObjectManager—The name of the Object Management server.
- Username—The username.
- Password—The user password.

General

In the General section, you can modify where to locate the Siebel Repository, the Login Time Out interval. From the Configuration Settings Editor main dialog:

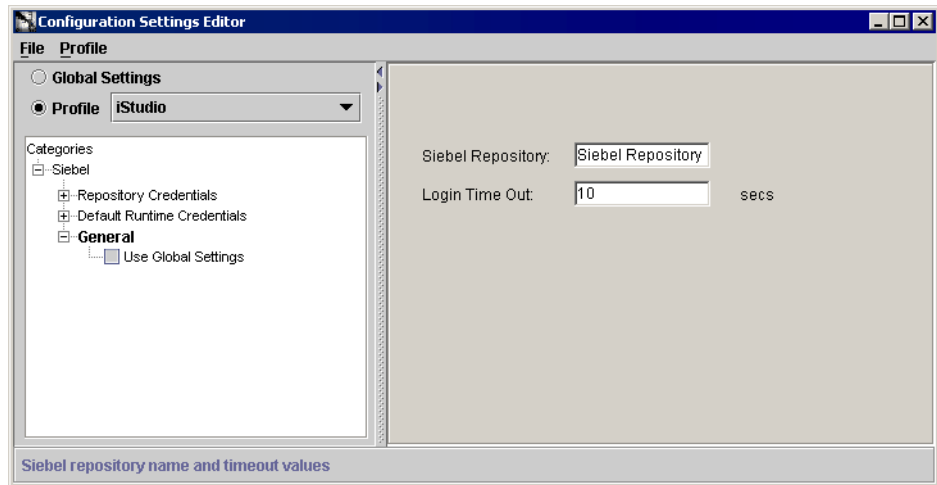
1. Click to expand **General**.
2. Click to uncheck **Use Global Settings**.
3. Click **General**.

In the dialog to the right, you can edit three fields.

Siebel Repository

Enter the name of the repository where the Siebel Repository files are located. The Siebel Repository contains Business Service methods, Business Components, and typed method arguments.

Figure 5–3 Configuration Settings Editor Timeout



Login Time Out (in seconds)

This variable determines how long you have to wait for a response before the system timesout. This value must be set between 10 and 300 seconds. Networks with a higher load should have this timeout variable set to a higher limit to allow enough time for the request to process.

The purpose of having Login timeout is to keep the threads from deadlocking. If there is no response within the specified time, the system stops sending packets and issues a 'timeout'. At that point, you need to reinitiate a login.

Siebel Adapter Functionality

This chapter provides an overview of the functionality of the Siebel adapter. The following topics are discussed:

- [Control Flow Overview](#)
- [Siebel Interfaces](#)

Control Flow Overview

This section contains these topics:

- [Design Time](#)
- [Runtime](#)

Design Time

When the Siebel adapter starts up, using the login credentials and system information from iStudio, it creates and pools one or more instances of the Siebel Application object. The Siebel adapter presents lists of Business Object, Business Components and Business Services in the browser. When you click on one of these, its logical methods display, along with their signatures. You can import these methods in iStudio to create application views.

Runtime

Runtime is the time where the Siebel adapter is servicing requests or waiting for requests. The distributed application in the Siebel adapter is up and running. All calls into the Siebel Application are synchronous calls. The data for the calls get transformed into Java values that Siebel receives through the Siebel Java Data Bean.

Siebel Interfaces

The Siebel adapter allows you to browse and call Siebel Business Services (objects with methods) and Business Components (data sets which you can query and update).

Under the hood, all Siebel applications use Business Components and Business Services, the basic building blocks. Business components are representations of a set of one or more joined tables. They have certain table-like behaviors such as the ability to query a set of records. Business Services are effectively containers for code. Custom business services can be defined using the Siebel Tools, including defining a list of arguments and data types.

Business Services

Business Services are also called "services" in Siebel. The Siebel repository contains Business Service methods and typed method arguments. The Siebel adapter uses a query interface to find the repository objects for Business Services. All method arguments are passed into the Business Service through a single invocation API.

Arguments of all Siebel Simple Business Service types are supported. Custom "methods" (function calls) can be defined using the Siebel Tools, including defining a list of arguments and data types. Either Siebel Basic (much like Visual Basic) or Siebel eScript (similar to javascript or Java) is used to implement these functions.

For Business Components, each record can have fields of the following types:

- String—An unbound string (not fixed length).
- Number—A numeric string with a total length of 22. This total length includes any plus or minus sign, exponent symbols, decimal symbol, integer or decimal fraction digits.
- Date—A date in the format MM/DD/YYYY. For the months of January through September and days from the first through the ninth a single character is acceptable.
- Business services that have hierarchy of property set parameters are not currently supported.

Note: Limitation: Many business service method parameters are not strongly typed in Siebel meaning that when they are created you can choose to type a parameter as a string even though internally it is used as a Number or Date. If such a business service method is found, you can remedy this by changing the parameter type to match how the parameter is actually used. See your Siebel system administrator or Siebel Tools expert for more information.

Business Objects and Business Components

Business Components are representations of a set of one or more joined tables. They have certain "table-like" behaviors such as the ability to retrieve a set of records. Business Objects are containers for Business Components.

For Business Components:

- Their data types are all complex.
- These complex data types are record sets.
- These are presented as Java vectors or arrays containing Java objects with the values.

For business components, each record can have fields of the following types:

- DTYPE_BOOL—Boolean T/F.

- DTYPE_CURRENCY—A numeric string (has no currency symbols, but handles the locale-based decimal symbol, and +/- prefix/suffix).
- DTYPE_DATE—A date in the Siebel adapter's Date format.
- DTYPE_DATETIME—A datetime in the Siebel adapter's DateTime format.
- DTYPE_TIME—A time in the Siebel adapter's Time format.
- DTYPE_INTEGER—A 32-bit integer.
- DTYPE_NOTE—An unbounded text field.
- DTYPE_NUMBER—Similar to DTYPE_CURRENCY but not used as currency in Siebel.
- DTYPE_PHONE—A string representing a phone number. It can be of two formats: +<any string>, or US area code and number: XXX-XXX-XXXX.
- DTYPE_TEXT (includes length of the string)—Strings of fixed length. The exact length of a DTYPE_TEXT field is visible in the Siebel adapter browser.

Unformatted DTYPE_PHONE Values

All phone numbers should be submitted as unformatted. A formatted phone number looks like:

```
(514) 332-6430 x909
```

An unformatted phone number looks like:

```
5143326430x909
```

A formatted value for a record field of type DTYPE_PHONE is a value whose format matches the locale for which the Siebel application server has been configured.

The Siebel adapter expects unformatted phone number values for inserts and updates.

If you submit a formatted phone number value for any business component phone field (of type DTYPE_PHONE), the insert/update will succeed for the first record, but all subsequent queries to retrieve that record fail with a seemingly unrelated error message back from Siebel. If you insert multiple records in a single Insert call, the first record insertion succeeds and the rest fail.

The following is the log exception if formatted phone number values are submitted:

Exception occurred:

Source: Siebel

Error Code: 65538 (0x10002)

Cause: Siebel://exception=SBLException (Unique ID <none>)

E-SBL0041: Call to business component Insert failed- Check record at index 1- Siebel error msg- Failed to retrieve the new record that was inserted. Contact Siebel Administrator Could not move the record pointer to first record. The insert operation failed at index 1- Siebel error code- 65538- Siebel record index- 1

Index

A

account

- child components, 4-4
- parent components, 4-4

advanced queuing adapter

- configuration, 2-4
- installation, 2-2

application parameter, 2-5

B

business address, 4-6

- parent components, 4-6

business objects and components, 6-3

business services, 6-2

C

component relationships, 4-2

components

- most commonly used, 4-2

configuration, 2-4

- adapter.ini initialization parameter file, 2-6
- advanced queuing adapter parameters, 2-11
- agent connection parameters, 2-7
- directories, 2-5
- executable files, 2-4
- files, 2-5
- hub.ini, 2-6
- siebel application server, 1-3

configuration editor, 5-2

contact, 4-6

- child components, 4-9

parent components, 4-8

control flow

- overview, 6-2

creating

- procedure, 3-2
- subscribed event, 3-8

D

design time, 6-2

DTYPE_PHONE

- unformatted values, 6-4

E

event

- creating subscribed, 3-8

F

frequently asked questions, 4-30

G

general, 5-4

I

installation, 2-2

- preinstallation tasks, 2-2
- required files, 1-3
- siebel, 2-2
- steps, 2-2
- supported platforms, 1-2

supported systems, 1-2
internal product, 4-9

L

login time out, 5-5

M

multi-valued fields, 4-28

O

opportunity, 4-14
 child components, 4-16
 parent components, 4-16
opportunity orders, 4-17
opportunity product, 4-19
 child components, 4-20
 parent components, 4-19
order entry
 other components interact, 4-20
order entry line items, 4-23
 child components, 4-24
 parent components, 4-23
order entry orders, 4-20
 child components, 4-22
 parent components, 4-22

P

price list, 4-10
 child components, 4-11
 parent components, 4-11
price list item, 4-11
procedure
 create implemented, 3-2

Q

quote, 4-12
 child components, 4-13
 parent components, 4-13
quote item, 4-14
 parent components, 4-14

R

rma, 4-24
 child components, 4-25
 parent components, 4-25
run time, 6-2

S

service request, 4-25
 child components, 4-27
 parent components, 4-27
siebel
 interfaces, 3-2
 what is, 1-2
siebel application server
 configuration, 1-3
siebel interfaces, 6-2
siebel login, 5-3
siebel repository, 5-5