



Implementing Siebel eBusiness Applications on DB2 UDB for z/OS and OS/390

Version 7.7, Rev. B
May 2005

Siebel Systems, Inc., 2207 Bridgepointe Parkway, San Mateo, CA 94404
Copyright © 2005 Siebel Systems, Inc.
All rights reserved.
Printed in the United States of America

No part of this publication may be stored in a retrieval system, transmitted, or reproduced in any way, including but not limited to photocopy, photographic, magnetic, or other record, without the prior agreement and written permission of Siebel Systems, Inc.

Siebel, the Siebel logo, UAN, Universal Application Network, Siebel CRM OnDemand, TrickleSync, Universal Agent, and other Siebel names referenced herein are trademarks of Siebel Systems, Inc., and may be registered in certain jurisdictions.

Other product names, designations, logos, and symbols may be trademarks or registered trademarks of their respective owners.

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

U.S. GOVERNMENT RESTRICTED RIGHTS. Programs, Ancillary Programs and Documentation, delivered subject to the Department of Defense Federal Acquisition Regulation Supplement, are "commercial computer software" as set forth in DFARS 227.7202, Commercial Computer Software and Commercial Computer Software Documentation, and as such, any use, duplication and disclosure of the Programs, Ancillary Programs and Documentation shall be subject to the restrictions contained in the applicable Siebel license agreement. All other use, duplication and disclosure of the Programs, Ancillary Programs and Documentation by the U.S. Government shall be subject to the applicable Siebel license agreement and the restrictions contained in subsection (c) of FAR 52.227-19, Commercial Computer Software - Restricted Rights (June 1987), or FAR 52.227-14, Rights in Data—General, including Alternate III (June 1987), as applicable. Contractor/licensor is Siebel Systems, Inc., 2207 Bridgepointe Parkway, San Mateo, CA 94404.

Proprietary Information

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

Contents

Chapter 1: What's New in This Release

Chapter 2: Preparing to Deploy Siebel on DB2 for z/OS and OS/390

Chapter 3: Security Concepts for z/OS and OS/390

- About Siebel Application Data Security 15
- Operating System Security 16
- Password Change and Expiration 16
- Changing Passwords from Remote Clients 16
- z/OS and OS/390 Security 17
 - Using a Secondary Authorization ID 17
 - About Using an External Security Adapter 17
- Data Transmission Security for Siebel Clients 18
 - Roles and Permissions Used to Connect to DB2 18
 - Granting SELECT Authority to Access the DB2 Catalog 19
- Required Authorizations 21

Chapter 4: Preparing for Implementation on the DB2 Host

- About System Connectivity Architecture 25
- Connecting to the Database Using DB2 Connect 26
- Installing Required IBM Fix Packs 27
- Setting Up DB2 Connect 28
 - Configuring DB2 Connect EE or UE to Support Siebel Web Client 28
 - Configuring DB2 Connect EE to Support Siebel Dedicated Web Client 29
 - Configuring DB2 Connect PE to Support Siebel Dedicated Web Client 29
- Additional Postinstallation Steps for DB2 Connect 30
 - Defining a Database Alias and Testing a Connection 30
 - Binding the DB2 Connect Packages 32
 - Configuring DB2 Connect for DB2 Connect Enterprise Edition Only 32
- Setting Up the Subsystem 34

About Unicode Conversion Tables	35
Choosing the Database CCSID	35
Using the Data Distribution Facility and Workload Manager	36
DSNZPARM Parameters for Setting Up the Subsystem	36
Estimating the Storage Space Required	39
Allocating Space for Buffer Pools and Storage Groups	40
Planning the Distribution of Database Objects	40
Estimating the Number of Database Objects You Need	41

Chapter 5: Configuring the Database Layout

About the Siebel Database Server File Installation	43
About the Siebel Storage Control File Templates	44
About Siebel Objects	45
About Substituting Values Using Symbolic Variables	45
About Default Objects in Storage Control Files	46
About Buffer Pools Used in Storage Control File Templates	47
About Using Storage Groups	47
About Databases in Storage Control Files	48
About Tables in Storage Control Files	48
About Table Spaces in Storage Control Files	49
About Object Naming Conventions	50
About Table Space Naming Conventions	50
About Creating or Modifying the Database Layout	52
About the Siebel Database Storage Configurator	53

Chapter 6: Understanding Siebel Partitioning

About Siebel Partitioning	55
Partitioning Strategy for Storage Control Files	56
About Configuring Siebel Templates for Partitioning	57
About Partitioning Tables	58
About Methods Used for Partitioning	58
Scenario for Partitioning	59
About Changing Partitioning Keys	62
Strategies for Special Types of Tables	63
About the Siebel Remote Transaction Logging Table	63
Partitioning S_ESCL_REQ and S_ESCL_LOG Tables	64
Partitioning EIM Tables for Performance	64

About EIM Considerations for Table Space Partitioning	65
About Prepartitioned Siebel Tables	65

Chapter 7: Installing the Siebel Database Server Files at Midtier

About File Installation at Midtier	71
About the Siebel Database Server	72
Preinstallation Tasks for the Siebel Database Server	72
Siebel Database Server Software Installation	73
Installing the Database Server Software Under Windows	73
Installing the Database Server Software Under UNIX	76
Reviewing the Software Installation for DB2 UDB for z/OS and OS/390	80

Chapter 8: Siebel Schema Installation on the DB2 Host

About Installing the Siebel Schema on the DB2 Host	83
Installing the Siebel Schema	84
Performing a Standard Installation	84
Performing a Standard Installation Under Windows	85
Performing a Standard Installation Under UNIX	90
Applying DDL on the DB2 Host	95
Preparing the Database Layout	101
Extracting a Storage Control File from the DB2 Catalog	102
Validating the Storage Control File	105
Using the Siebel Database Storage Configurator	108
Configuring a Siebel Storage Control File	109
Creating and Executing the DDL	113
Moving DDL Files to the S/390 or zSeries Machine	113
Splitting the DDL Job	113
Performing a Custom Installation	114
Performing a Custom Installation Under Windows	115
Reviewing the Log Files for Windows	117
Performing a Custom Installation Under UNIX	118
Reviewing the Log Files for UNIX	119
Rerunning the Installation	120

Chapter 9: Siebel Repository Import, Export, and Postinstallation Tasks

- About the Database Server Installation and Configuration 121
- Importing and Exporting the Siebel Repository 122
- Reviewing the Log Files for Repository Import 128
 - About Acceptable Errors on the errors.rtf File 128
 - Troubleshooting the Repository Import 129
- Postinstallation Tasks for the Siebel Repository 129
 - Granting Table Privileges 129
 - Loading Statistics 129
 - Validating the Siebel Schema 130
 - Installing Multilingual Seed Data 132
 - Populating the Siebel File System 137

Chapter 10: Customizing a Development Environment

- About Customizing Your Development Environment 139
- Siebel Tools Considerations on DB2 UDB for z/OS and OS/390 140
 - Setting Siebel Tools Configuration File Parameters 140
 - Setting Database Options 140
 - Specifying Table Grouping File Names 141
 - Inactivating Unused Indexes 141
 - Reducing Field Lengths of Unique Indexes to Avoid Truncation 141
 - Reducing VARCHAR Field Lengths 141
 - LONG Columns on DB2 UDB for z/OS and OS/390 and Other DBMS Platforms 142
 - Changing LONGVARCHAR Datatype to VARCHAR Datatype 144
 - Using Character Large Objects (CLOBs) 145
- Converting Nonpartitioned Tables to Partitioned Tables 145
- Creating Custom Extensions to the Siebel Schema 146
 - About Schema Extension Tasks 147
 - Creating Storage Control File Objects for New Schema Objects 148
 - Applying Schema Extensions to the Development Database 148
- How to Spool DDL Generated by the Apply Button 150
- Applying Schema Extensions to the Target Database 150
 - Preparing the Target Database 150
 - Migrating Customizations from Development to the Target Database 151
 - Applying Schema Changes to the Target Database 155
 - Synchronizing Schema Definitions with the Existing Siebel Schema 157
- Cloning an z/OS and OS/390 Database 158

- Migrating Rules and Other Data 161
- Applying Schema Changes to Other Local Databases 162
- Customizing Applications Using Assignment and Workflow Rules 162

Chapter 11: Maintenance Considerations for zSeries

- About DB2 Statistics for Siebel eBusiness Applications 163
- About Reorganizing Table Spaces, Partitions, and Indexes 164
- About Cursor Close 165
 - About Cursor Operation in Unrestricted and Restricted Modes 165
 - Modifying the Cursor Close Alert Message 166
 - Cursor Close Ignored 167
 - About Cursor Configuration Parameter Values 167
- Setting Database Connection Pooling 170
- About Siebel Dynamic SQL 170
 - About Spooling SQL Within the Siebel Application 171
 - Enabling DB2 CLI Tracing 171
 - Enabling ODBC Tracing 172
 - Enabling SQL Tracing for Database Utilities 172
 - About SQL Tracing on the DB2 Host Using Tools Like DB2 Performance Monitor 172
 - About a RLF Error Message 173
- Using odbcsql to Submit SQL Statements 173
- About DB2 Dynamic Statement Caching 174
- About Tracing the Source of a Query 175
- About Coordinated Universal Time and DB2 UDB for z/OS and OS/390 176

Chapter 12: Migrating Data Using Siebel EIM

- Setting Up EIM for DB2 177
 - Choosing an Appropriate Partitioning Scheme 177
 - Considering the Effect of EIM on Table Partitioning 178
 - Planning Data Mapping 178
 - Choosing an Optimal EIM Batch Size 179
 - Choosing an Optimal Number of Parallel Streams 179
 - Optimizing DB2 Database Logging 179
 - Using the RUNSTATS Utility 180
 - Reviewing the SQLPROFILE Log 180
- Improving EIM Performance During the Import Process 180
 - Using the Default Column to Improve Throughput 181

Tuning Your EIM Import Process	181
Initial Load Considerations	182
Considerations for EIM Tables	183

Appendix A: Deployment Planning Worksheet

Master Worksheet	185
Team Lead Summary	185
DB2 Connect Information	186
Siebel Database Server Installation Information	186

Index

1

What's New in This Release

What's New in Implementing Siebel eBusiness Applications on DB2 UDB for z/OS and OS/390, Version 7.7, Rev. B

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

Table 1. New Features in Implementing Siebel eBusiness Applications on DB2 UDB for z/OS and OS/390, Version 7.7, Rev. B

Topic	Description
“Defining a Database Alias and Testing a Connection” on page 30	Amended topic. If database aliases are defined, the TXN1 SOLATION parameter must be added to the database alias entry in the db2cli.ini file, and must be set to have a value of 1.
“About Unicode Conversion Tables” on page 35	New topic. Unicode Conversion Services must be installed if Siebel eBusiness Applications are to communicate with a DB2 database that does not use Unicode character set.
“Using the Data Distribution Facility and Workload Manager” on page 36	Amended topic to clarify that Workload Manager must be installed and configured if the Siebel application stored procedures and user defined functions are to work.
“Reducing Field Lengths of Unique Indexes to Avoid Truncation” on page 141	Amended topic for clarification. Some indexes supplied by Siebel eBusiness Applications do not conform to the DB2 for z/OS v7 limitation of 255 characters.
“About DB2 Statistics for Siebel eBusiness Applications” on page 163	Amended topic. The RUNSTATS utility can be run on any table in Siebel 7.x.
“About Cursor Configuration Parameter Values” on page 167	Amended topic to describe how to set cursor configuration parameters for the Web Client and Siebel Server in release 7.7x.

What's New in Implementing Siebel eBusiness Applications on DB2 UDB for z/OS and OS/390, Version 7.7, Rev. A

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

Table 2. New Features in Implementing Siebel eBusiness Applications on DB2 UDB for z/OS and OS/390, Version 7.7, Rev. A

Topic	Description
"Planning the Distribution of Database Objects" on page 40	Information revised on the most commonly used or largest tables in the Siebel application.
"About Using an External Security Adapter" on page 17	The procedure for configuring LDAP deleted from guide. Users are referred to the <i>Security Guide for Siebel eBusiness Applications</i> for more information on configuring an external security adapter.
"Applying DDL on the DB2 Host" on page 95	Revised instructions for applying DDL on the DB2 host when you select the option Generate DDL Into Files during the standard installation process.
"Migrating Customizations from Development to the Target Database" on page 151	A change in the listing of files produced when migrating customizations from development to production.
"Applying Schema Changes to the Target Database" on page 155	New instructions added for applying the schema changes after performing a migration from development to production process.
"Calculating Space for Table Spaces and Indexes"	Calculating space for table spaces and indexes is not supported in this release. Section deleted from guide.
"Using the Create Alter Statements Function to Adjust Size"	The Create Alter Statements function is not supported in this release. Section deleted from guide.
"About the Siebel Database Storage Configurator" on page 53	The paragraph containing information about the dbconf.xls file that includes functions for calculating row count and available space based on row count was deleted. These functions are not supported in release 7.7.x.

What's New in Implementing Siebel eBusiness Applications on DB2 UDB for z/OS and OS/390, Version 7.7

Table 3 lists changes described in this version of the documentation to support Release 7.7 of the software.

Table 3. New Features in Implementing Siebel eBusiness Applications on DB2 UDB for z/OS and OS/390, Version 7.7

Topic	Description
“About the Siebel Storage Control File Templates” on page 44	<p>New Siebel Schema layout: one table per table space, one table space per database.</p> <p>As a result, the following changes occurred:</p> <ul style="list-style-type: none"> ■ Database name prefixes changed from six characters to four characters. ■ Siebel Systems ships three database storage control files. ■ Database storage layout from prior releases are no longer shipped.
“Applying DDL on the DB2 Host” on page 95	<ul style="list-style-type: none"> ■ New user-defined functions: <ul style="list-style-type: none"> ■ nextseq() generates sequential values for EIM Export. ■ exrate() currency aggregation functions. ■ DB2 options are supported by Siebel database utilities for CREATE statements.
“About a RLF Error Message” on page 173	DB2 RLF message provided to end users.
“About Tracing the Source of a Query” on page 175	Session ID tracking string can be used to identify the current connection.
“About Cursor Close” on page 165	Default values MaxCursorSize and PrefetchSize changed from 48 to 128.
“About Coordinated Universal Time and DB2 UDB for z/OS and OS/390” on page 176	Coordinated Universal Time (UTC) is supported on zSeries.

2

Preparing to Deploy Siebel on DB2 for z/OS and OS/390

Several items should be considered when preparing your implementation of Siebel eBusiness Applications using IBM DB2 UDB for z/OS and OS/390. These considerations are covered in-depth throughout this guide.

To prepare for Siebel deployment on DB2 for z/OS and OS/390

- 1** Review the prerequisites in the *System Requirements and Supported Platforms* on Siebel SupportWeb for your Siebel application version and make sure that all requirements are met. For the DB2 area, that includes:
 - Applying required OS maintenance as listed in the IBM informational APAR ii13487.
 - Installing a supported version of DB2 Connect and IBM Fix Pack.
 - Installing Unicode Conversion Services.
- 2** Review Siebel architecture with the DB2 for z/OS database server with Siebel Expert Services.
- 3** Decide on DB2-related architecture and other related choices:
 - Separate or shared DB2 subsystem for production.
 - ASCII or EBCDIC encoding scheme, code page selection.
 - Security scheme (native DB2 security or LDAP).
 - Whether you plan to implement UTC.
 - Remember that Siebel requires deployment of user-defined functions (UDFs) and stored procedures.
 - Understand and design the database storage layout for the Siebel schema.
 - Understand the Siebel Schema and review possible table space partitioning.
 - If batch processing through EIM is planned, review partitioning, so it will work for both batch and online activities.
 - Review and set up required and recommended DB2 system parameters (DSNZPARMs).
 - Configure the DB2 Data Distribution Facility and the z/Series Workload Manager (WLM).
 - After the installation is completed, set up Siebel parameters and environment variables as it suits your deployment.
 - Decide on backup and recovery mechanisms.
 - Decide on maintenance procedures, such as RUNSTATS and REORG.
 - Set up development, test, and other environments.
- 4** The implementation team should review the *Release Notes* and *Alerts* on Siebel SupportWeb to make sure that they are aware of known anomalies.

- 5 Have a project staff:
 - Dedicate DBAs to support Siebel deployment.
 - Establish good working relationships with mid-tier administrators, because some operations, such as installation or upgrade, require close cooperation between these groups of people.
 - DB2 DBAs using the Siebel application for the first time should be aware of the following because it contains:
 - Many objects, some of them unused.
 - Data types, mainly VARCHAR, that are used in the Siebel data model.
 - All dynamic SQL.
 - SQL that is generated by the Siebel application versus traditional SQL precisely coded by developers.

3

Security Concepts for z/OS and OS/390

This chapter discusses important security concepts as they relate to Siebel eBusiness Applications running on IBM's DB2 UDB for z/OS and OS/390. This chapter also describes the roles and permissions required to set up a connection between the Siebel Server and the database server residing on the DB2 host and to install the Siebel application on it.

For a discussion of Siebel architecture in general, including a discussion of supported client types, refer to the *Deployment Planning Guide* and the *Siebel Installation Guide* for the operating system you are using.

This chapter consists of the following topics:

- [“About Siebel Application Data Security” on page 15](#)
- [“Operating System Security” on page 16](#)
- [“Password Change and Expiration” on page 16](#)
- [“Changing Passwords from Remote Clients” on page 16](#)
- [“z/OS and OS/390 Security” on page 17](#)
- [“Data Transmission Security for Siebel Clients” on page 18](#)
- [“Required Authorizations” on page 21](#)

About Siebel Application Data Security

All users must provide a user name and password to connect to Siebel eBusiness Applications, regardless of whether they access them through the Siebel Dedicated Web Client, the Siebel Mobile Web Client, or the Siebel Web Client.

Each user name must be preregistered within the Siebel application and associated with a unique record. This unique record determines a user's visibility to data by association with Positions and Responsibilities.

- Responsibilities determine which screens and views are available to an employee.
- Positions determine which data rows appear in those views, and which data rows are synchronized to a Mobile Web Client's local database.

Siebel eBusiness Applications allow you to control user access to information from within the application. Because the need to know specific information is usually related to a particular responsibility, data access is defined by responsibility rather than by user.

Attachments that are not stored in the database are stored in a compressed, encrypted format on the Siebel File Server. Attachments are linked to data rows, and access to attachments is therefore restricted by a user's responsibilities and position.

For more information on these topics, refer to *Applications Administration Guide* and *Security Guide for Siebel eBusiness Applications*.

Operating System Security

The DB2 Connect middleware passes the user name and password used to access Siebel eBusiness Applications to DB2 to establish a connection. The user name and password can be user credentials supplied by the external security adapter if an external security adapter is being used.

All user names and passwords passed through DB2 Connect must have been preregistered by the Security Administrator (or by someone who has been delegated this authority) with the z/OS or OS/390 security package (such as RACF, ACF/2, or TOP SECRET) in use.

The user name can be associated by the z/OS or OS/390 Security Administrator with a secondary authorization ID (also known as the Security Group ID on the [“Deployment Planning Worksheet” on page 185](#)) within the z/OS or OS/390 security package to simplify database security privilege administration.

DB2 Connect supports password encryption on the DB2 UDB for z/OS and OS/390 platform.

NOTE: Siebel user IDs and passwords for DB2 are case-insensitive.

Password Change and Expiration

Because of security constraints, zSeries customers may have user passwords periodically expire.

- **When a password has expired.** Siebel eBusiness Applications support the ability to change expired passwords from the login window without administrative intervention.
- **When a password has not yet expired.** If a password has not yet expired, passwords can be changed through the Change Password client function.

Changing Passwords from Remote Clients

In order to change z/OS and OS/390 passwords from remote client machines, set the DB2 extended security option (EXTSEC) to YES. The default DB2 value for extended security is NO. There are two ways to set the extended security option to YES:

- In the DSNTIPR installation panel, change EXTENDED SECURITY to YES.
- In the DSN6SYSP macro, set the DSNZPARM EXTSEC=YES.

Setting the extended security option to YES enables the following two functions:

- Users on client workstations can change their z/OS and OS/390 passwords without signing onto TSO.
- Siebel eBusiness Applications receive descriptive error codes generated by DB2 when security violations occur.

z/OS and OS/390 Security

By default, DB2 restricts access to database resources unless privileges are specifically granted. A complete description of DB2 security is available in the vendor documentation on the IBM website. Access to database resources is established using one or more of the following authorization IDs:

- **Primary authorization ID.** The user ID used to log into DB2. All users will have a primary authorization ID.
- **Secondary authorization ID.** Generally known as a group ID. Siebel applications effectively use secondary authorization IDs that can be enabled through the DB2 installation exit, DSN3@ATH. IBM provides a sample exit for setting secondary authorization IDs.
- **Package owners.** The ID of the owner of a DB2 Connect package. It is used for authorization purposes if the package is bound with DYNAMICRULES(BIND) on the BIND command. For more information on DB2 Connect Packages, see an IBM Corporation Redbook on this subject.

Using a Secondary Authorization ID

The use of a secondary authorization ID significantly reduces administrative tasks associated with database security. The administrator grants privileges only once to a secondary authorization ID rather than to each Siebel eBusiness Applications user.

NOTE: During the Siebel Database Server installation, on the Installation and Configuration Screen, you will be prompted for a Security Group ID/Grantee. This is the same as a secondary authorization ID.

The Siebel installation process allows installers to specify a secondary authorization ID for client access with the default group of SSEROLE. The installation process generates the appropriate SQL grant statements for that group to allow INSERT, UPDATE, SELECT, and DELETE to application tables. Furthermore, that same group is specified in a SET CURRENT SQLID statement so that reuse of the statement cache is maximized. Therefore, it is important that the selected group be among the list of secondary authorization IDs for all users of the applications.

Grant statements for additional secondary authorization IDs. Secondary authorization IDs must be created separately. Siebel applications include the grantstat.sql script to generate grant statements to grant access to interface tables. For a discussion of the grantstat.sql script, see [“Granting Table Privileges” on page 129](#).

The grant statements must be executed by either the table owner or by users with DBADM or SYSADM privileges.

To disable a grant, issue a revoke statement.

About Using an External Security Adapter

An external security adapter is an interface that lets you use an external system to authenticate users. For example, you might employ an LDAP (Lightweight Data Access Protocol) repository, a protocol for storing and retrieving directory-related information that includes authentication services. LDAP can reside on the mainframe.

Using an external security adapter makes administration easier, because you do not have to create an account for each Siebel application user on the DB2 host. You can instead create a few generic database accounts that are used by multiple Siebel users.

When users log onto the Siebel application, the external security adapter validates user names, passwords, user roles, and database credentials against the information in the external system. If the external security adapter finds a match, it retrieves a generic set of user credentials (username and password) that supply access to the database. For LDAP, the generic set of user credentials can be the same for every user, if desired.

For more information about implementing an external security adapter, see the *Security Guide for Siebel eBusiness Applications*.

Data Transmission Security for Siebel Clients

Siebel Web Clients and Mobile Web Clients access Siebel eBusiness Applications and, in the latter case, can synchronize with the corporate database over the Internet. To provide data transmission security for these situations, Siebel Systems supports compression and encryption of data between these clients and the Siebel Application Object Manager and Remote Manager processes.

For more information about this topic, refer to the *Siebel Installation Guide* for the operating system you are using.

DB2 Connect also supports password encryption on the DB2 UDB for z/OS and OS/390 platform. For more information about this topic, refer to *IBM DB2 Universal Database Command Reference*, IBM Redbooks, and other IBM documentation.

Roles and Permissions Used to Connect to DB2

The following roles and permissions are used to connect to DB2 and to install Siebel eBusiness Applications on a DB2 database:

- SYSADM
- DBADM
- CREATEDBA

SYSADM Privileges Used for Connecting to DB2

A DB2 subsystem is a prerequisite for installing Siebel eBusiness Applications. While installation does not require the use of an ID with SYSADM privileges, the creation of some underlying DB2 resources might. For detailed information on setting up a DB2 subsystem for Siebel eBusiness Applications, see [Chapter 4, "Preparing for Implementation on the DB2 Host."](#)

Functions that require SYSADM authority and that are necessary for installing Siebel eBusiness Applications on DB2 UDB for z/OS and OS/390 include:

- Allocating and accessing buffer pools

- Allocating and accessing storage groups
- Granting CREATEDBA or DBADM to the Siebel user ID used for the Siebel database installation
- User Defined Functions and stored procedures

DBADM/CREATEDBA Privileges Used for Connecting to DB2

Database Server installation requires access similar to that of DBADM; installers must be able to create Siebel objects and access the necessary utilities. Therefore, it is recommended that you grant CREATEDBA privileges to the primary or secondary authorization IDs that will be used to perform the database server installation.

Granting SELECT Authority to Access the DB2 Catalog

Siebel eBusiness Applications access the DB2 catalog to validate installation inputs.

To grant appropriate users access privileges to the DB2 catalog, the system administrator must grant SELECT authority on certain catalog tables to users of the Siebel Database and to users of the installation or upgrade utility.

- Siebel Database users (also known as privileged users who make changes to the Siebel database) require SELECT authority for SYSIBM.SYSTABLES.
- Database Server installation or upgrade users require SELECT authority for the following tables:
 - SYSIBM.SYSAUXRELS
 - SYSIBM.SYSCOLUMNS
 - SYSIBM.SYSDATABASE
 - SYSIBM.SYSINDEXES
 - SYSIBM.SYSKEYS
 - SYSIBM.SYSROUTINES
 - SYSIBM.SYSSTOGROUP
 - SYSIBM.SYSTABLESPACE
 - SYSIBM.SYSTABLES
 - SYSIBM.SYSTABLEPART
 - SYSIBM.SYSTRI G GERS

To grant SELECT authority to access the DB2 Catalog

- Use the following command:

```
GRANT AUTHORITY_TYPE ON TABLE TABLENAME TO USER;
```

For example, to grant SELECT authority on the table SYSIBM.SYSTABLES to SSEROLE, use the following command:

```
GRANT SELECT ON TABLE SYSIBM.SYSTABLES TO SSEROLE;
```

Granting UPDATE Authority to Update DB2 Catalog Statistics

NOTE: Updating DB2 catalog statistics is optional.

To give appropriate users the authority to update catalog statistics using the loadstat.sql script, the system administrator must grant UPDATE authority to installation or upgrade users for the following tables:

SYSIBM.SYSCOLUMNS

SYSIBM.SYSINDEXES

SYSIBM.SYSINDEXSTATS

SYSIBM.SYSKEYS

SYSIBM.SYSTABLES

SYSIBM.SYSTABLESPACE

SYSIBM.SYSTABSTATS

The loadstat.sql script is provided in the dbsrvr/DB2390 directory, which is created during the database server installation. For information on using this script, see [“Loading Statistics” on page 129](#).

Required Authorizations

Table 4 lists the authorizations that are required by a DB2 database administrator or system programmer to implement Siebel eBusiness Applications on DB2 UDB for z/OS and OS/390.

Table 4. Authorizations Needed by DB2 Database Administrators

Task	Authorization Required	Task Command Example
Alter a buffer pool.	SYSADM, SYSCTRL, SYSOPR	ALTER BUFFERPOOL (BP32K1) VPSIZE(4000);
Grant use of a buffer pool.	SYSADM, SYSCTRL	GRANT USE OF BUFFERPOOL BP32K1 TO PUBLIC;
Grant CREATEIN for triggers.	SYSADM, SYSCTRL	GRANT CREATEIN ON SCHEMA SIEBTO;
Create a storage group.	SYSADM, SYSCTRL	CREATE STOGROUP SIEBEL VOLUMES('*') VCAT SIEBEL;
Grant use of a storage group.	SYSADM, SYSCTRL	GRANT USE OF STOGROUP SIEBEL TO PUBLIC;
Grant CREATEDBA and DBADM authority.	SYSADM, SYSCTRL	GRANT CREATEDBA TO SIEBTO;
Create a database.	SYSADM, SYSCTRL, CREATEDBA, CREATEDBC	SET CURRENT SQLID='SIEBTO'; CREATE DATABASE SIDB0001 CCSID ASCII BUFFERPOOL BP1;
Alter a table space.	DBADM, SYSADM, SYSCTRL	ALTER TABLESPACE D0010002.H0677000 BUFFERPOOL BP16K0;
Create a table space.	SYSADM, SYSCTRL, DBADM, DBCTRL, DBMAINT, CREATETS	SET CURRENT SQLID='SIEBTO'; CREATE TABLESPACE H1004000 IN SIDB1004 USING STOGROUP SIEBEL PRIQTY 720 SECQTY 720 BUFFERPOOL BP32K1 SEGSIZE 64 COMPRESS YES LOCKSIZE PAGE;
Modify DB2 Connect package (if package already exists).	DBADM, SYSADM, BIND privilege on the package, ALTERIN privilege on the schema	BIND c:\sqllib\bnd\@ddcsmvs.lst COLLECTION SIEBEL BLOCKING ALL DYNAMICRULES BIND OWNER SIEBTO QUALIFIER SIEBTO SQLERROR CONTINUE

Table 4. Authorizations Needed by DB2 Database Administrators

Task	Authorization Required	Task Command Example
Add DB2 Connect package (if a package does not already exist).	DBADM, SYSADM, BINDADD privilege, and IMPLICIT_SCHEMA authority on the database if the schema name does not exist CREATIN privilege on the schema if the schema name of the package exists	BIND c:\sqllib\bnd\@ddcsmvs.lst COLLECTION SIEBEL BLOCKING ALL DYNAMICRULES BIND OWNER SIEBTO QUALIFIER SIEBTO SQLERROR CONTINUE
Alter a table.	DBADM, SYSADM, SYSCTRL	ALTER TABLE SIEBTO.S_CONTACT ALTER COLUMN COMMENTS SET DATA TYPE VARCHAR (100);
Create a table.	SYSADM, SYSCTRL, DBADM, DBCTRL, DBMAINT, CREATETAB	SET CURRENT SQLID='SIEBTO'; CREATE TABLE S_SSA_ID,...)IN SIDB1932.H1932000;
Alter an index.	DBADM, SYSADM, SYSCTRL	ALTER INDEX SIEBTO.S_CONTACT_U1 BUFFERPOOL BP30;
Create an index.	SYSADM, SYSCTRL, DBADM, DBCTRL	SET CURRENT SQLID='SIEBTO'; CREATE UNIQUE INDEX S_SSA_ID_P1 ON S_SSA_ID(ROW_ID)USING STOGROUP SIEBEL PRIQTY 720 SECQTY 720 DEFINE NO CLOSE YES PCTFREE 30 BUFFERPOOL BP2;
Grant CREATE or PACKADM for stored procedures.	SYSADM, SYSCTRL	GRANT CREATE ON COLLECTION SIEBINST TO SIEBDBA;
Grant BINDADD.	SYSADM, SYSCTRL	GRANT BINDADD TO SIEBDBA;
Grant SELECT on catalog tables.	SYSADM, SYSCTRL	GRANT SELECT ON SYSIBM.SYSTABLES TO SIEBTO;

Table 5 lists authorizations that might be needed by users of Siebel eBusiness Applications. Your enterprise might have unique role names it assigns with the authorities listed below. Therefore, the sample role names are offered as examples only.

Table 5. Authorizations Needed by Siebel Application Users

Task	Role	Authorization Required	Task Command Example
Performing the following actions on Siebel tables: ■ Delete ■ Insert ■ Select ■ Update	Siebel group ID (for example, SSEROLE group)	Table privileges granted automatically as part of installation	GRANT ALL ON TABLE S_SSA_ID TO SSEROLE;
Setting the current SQL ID	Schema qualifier group or individual ID	Owens the schema objects (created by DBADM) that will be used during the installation of the Siebel product	SET CURRENT SQLID='SIEBTO' ;
Performing server functions such as: ■ Runstats ■ Generate triggers	Siebel administrator group	■ Member of the generic user role ■ Actual member of the RACF (or other security package) group, selected to act as administrator	SIEBDBA

4

Preparing for Implementation on the DB2 Host

This chapter is intended primarily for the z/OS and OS/390 system programmer and DBA who must prepare for installing the Siebel Schema on the DB2 host.

This chapter consists of the following topics:

- [“About System Connectivity Architecture”](#)
- [“Connecting to the Database Using DB2 Connect” on page 26](#)
- [“Installing Required IBM Fix Packs” on page 27](#)
- [“Additional Postinstallation Steps for DB2 Connect” on page 30](#)
- [“Setting Up the Subsystem” on page 34](#)
- [“Estimating the Storage Space Required” on page 39](#)
- [“Allocating Space for Buffer Pools and Storage Groups” on page 40](#)
- [“Planning the Distribution of Database Objects” on page 40](#)
- [“Estimating the Number of Database Objects You Need” on page 41](#)

About System Connectivity Architecture

The Siebel eBusiness Applications communicate with IBM DB2 UDB for z/OS and OS/390 through IBM DB2 Connect middleware.

The three editions of DB2 Connect are:

- **DB2 Connect Enterprise Edition (EE).** Supports database connectivity for users running Siebel Web Client when communicating with the Siebel Server. Optionally, this edition can support database connectivity for multiple individual users running Siebel Dedicated Web Client. This edition is installed on a midtier server such as a Siebel Server machine.
- **DB2 Connect Unlimited Edition (UE).** Provides the functionality of DB2 Connect Enterprise Edition but is priced differently.
- **DB2 Connect Personal Edition (PE).** Supports database connectivity for an individual user running Siebel Dedicated Web Client on a workstation. This edition is installed on a user’s local workstation.

DB2 Connect EE or UE can reside either on the same machine as Siebel Server or on common connection gateway machines. (Do not confuse the concept of a *gateway* machine with the Siebel Gateway Name Server. DB2 Connect EE or UE is used to broker the connections to DB2 for multiple database clients (Siebel Servers or Dedicated Web Clients). See [Table 6](#) for DB2 Connect configurations.

NOTE: If you deploy DB2 Connect EE or UE with Siebel Dedicated Web Client, you must also install a DB2 Connect run-time component on your Siebel Dedicated Web Client machines. The DB2 Runtime *must* be installed on the Siebel Server if it does not co-exist with DB2 Connect.

Siebel Servers and Siebel Dedicated Web Clients communicate with DB2 Connect through TCP/IP. (DB2 Connect also supports communication through SNA, but SNA currently does not support connection pooling. For this reason, it is recommended that you use TCP/IP.)

Connecting to the Database Using DB2 Connect

You install the Siebel Database Server on DB2 UDB for z/OS and OS/390 using either direct execution methods or the Database Server configuration wizard on a client Siebel Server. (Installation methods are discussed in [Chapter 8, "Siebel Schema Installation on the DB2 Host."](#)) Subsequently, you will execute SQL on DB2.

Configuration Options for DB2 Connect

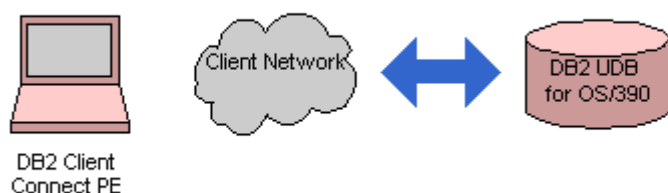
If you are using DB2 Connect, your configuration options depend on whether you are deploying Siebel eBusiness Applications on a Web Client or on a Dedicated Web Client.

Table 6. DB2 Connect Configurations

DB2 Connect Edition	Siebel eBusiness Applications Deployed on	Install and Run DB2 Connect on
Enterprise Edition (EE or UE)	Siebel Web Client	Siebel Server machine or another machine on midtier.
Enterprise Edition (EE or UE)	Siebel Dedicated Web Client	Any machine on midtier.
Personal Edition (PE)	Siebel Dedicated Web Client	Each workstation.

Figure 1 illustrates configurations with DB2 Connect Personal Edition (PE) and DB2 Connect Enterprise Edition (EE). The configuration you choose largely depends on the types of Siebel clients your enterprise supports.

DB2 Connect Personal Edition Runs on the Workstation Only



DB2 Connect Enterprise Edition Runs on the Server(s)

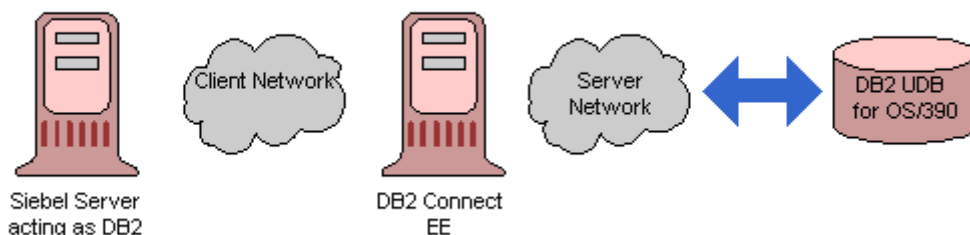


Figure 1. DB2 Connect Enterprise Edition and Personal Edition

For detailed information about DB2 Connect, refer to the IBM documentation.

Installing Required IBM Fix Packs

After you have installed DB2 Connect, you must install any required IBM Fix Packs. Not all Fix Packs are certified to run with Siebel applications. To find out which Fix Packs are certified, see the *System Requirements and Supported Platforms* on Siebel SupportWeb or the vendor documentation on the IBM Web site. For more information, refer to IBM documentation for installing the Fix Packs for your environment.

NOTE: When using the UNIX operating system, if a Fix Pack is installed after the creation of the DB2 database instance, this instance must be updated using the `db2iupdt` command.

You can verify the currently installed version and Fix Pack by running the command `db2level` from a DB2 Command Window. In a UNIX environment, this command is run from a UNIX shell after sourcing the appropriate `db2profile`. Make note of the information and compare the Informational tokens from the output against the *System Requirements and Supported Platforms* on Siebel SupportWeb or the IBM URL <ftp://ftp.software.ibm.com/ps/products/db2/fixes/english-us/siebel/siebel7/Siebel7CertifiedFixpaks.txt>.

In addition to Fix Pack information the *System Requirements and Supported Platforms* on Siebel SupportWeb contains any other installation prerequisites.

Setting Up DB2 Connect

To set up DB2 Connect, follow the instructions below that apply to your chosen DB2 Connect configuration.

- **Define a database alias.** Define a database alias with the proper connection information for your database. This alias is the connect string that you specify when you install Siebel Server. Record the connect string in the copy you made of [Appendix A, “Deployment Planning Worksheet.”](#)

To define your database alias, use the DB2 Client Configuration Assistant or the DB2 Command Line Processor. For more information, refer to the vendor documentation on the IBM Web site.

- **Enable ODBC.** Enable ODBC to point to DB2 UDB for z/OS and OS/390.

To enable ODBC to point to DB2, use the DB2 Client Configuration Assistant’s BIND option to bind the CLI/ODBC Support packages. Make sure that the authid has enough authority to bind these packages (BINDADD).

- **Install TCP/IP and FTP.** For instructions on installing TCP/IP, refer to the vendor documentation on the IBM Web site.

Configuring DB2 Connect EE or UE to Support Siebel Web Client

Using DB2 Connect EE or UE to support the Siebel Web Client involves:

- Siebel Server, running Application Object Managers to support Siebel Web Client. In some cases, DB2 Connect EE is also installed on this machine.
- Optionally, an additional computer on which you install IBM’s DB2 Connect EE, to act as a gateway to the database.
- DB2 UDB for z/OS and OS/390.
- Individual workstations running Siebel Web Client.

This approach is illustrated in the lower part of [Figure 1 on page 27](#).

Perform the following steps to configure DB2 Connect EE for a server machine.

To connect to DB2 using DB2 Connect EE

- 1 Using vendor instructions, install DB2 Connect EE on a server machine.

The DB2 Connect EE Server machine functions as a DB2 UDB server, with protocol support for DB2 UDB for z/OS and OS/390.

You can also add DB2 Connect to an existing server with DB2 already installed.

- 2 On the DB2 Connect machine, upgrade DB2 Connect to the appropriate Fix Pack.

For information regarding the supported Fix Pack version, refer to *System Requirements and Supported Platforms* on Siebel SupportWeb.

- 3 On the DB2 Connect EE Server machine, catalog your DB2 z/OS and OS/390 database, as appropriate, using the DB2 Client Configuration Assistant or the DB2 Command Line Processor.
After installation, use a standard DB2 UDB client to access the DB2 Connect EE Server.

Configuring DB2 Connect EE to Support Siebel Dedicated Web Client

Using DB2 Connect EE to support the Siebel Dedicated Web Client involves:

- A computer on which you install IBM's DB2 Connect EE, to act as a gateway to the database. This computer might be a Siebel Server machine or a separate machine.
- DB2 UDB for z/OS and OS/390.
- Siebel Dedicated Web Client workstations.

This configuration enables a high volume of concurrent network transactions between Siebel Dedicated Web Clients and the DB2 Connect EE Server machine. This approach combines elements from both scenarios shown in [Figure 1 on page 27](#).

To connect to DB2 using DB2 Connect EE

- 1 Using vendor instructions, install DB2 Connect EE on a gateway machine.
The DB2 Connect EE Server machine functions as a DB2 UDB server, with protocol support for DB2 UDB for z/OS and OS/390.
- 2 On the DB2 Connect EE Server, upgrade DB2 Connect EE to the appropriate Fix Pack.
For information regarding the supported Fix Pack version, refer to *System Requirements and Supported Platforms* on Siebel SupportWeb.
- 3 On the DB2 Connect EE Server machine, catalog your DB2 z/OS or OS/390 database, as appropriate, using the DB2 Client Configuration Assistant or the DB2 Command Line Processor.
After installation, use a standard DB2 UDB client to access the DB2 Connect EE Server.

Configuring DB2 Connect PE to Support Siebel Dedicated Web Client

Using DB2 Connect PE to support the Siebel Dedicated Web Client involves:

- Siebel Dedicated Web Client workstations on which you install IBM's DB2 Connect Personal Edition (PE), to act as a gateway to the database
- DB2 UDB for z/OS and OS/390

With this approach, users on a Siebel Dedicated Web Client connect directly to DB2. See the upper part of [Figure 1 on page 27](#).

To connect to DB2 using DB2 Connect PE

- 1** Install DB2 Connect PE on each Siebel Dedicated Web Client workstation machine.
- 2** On each machine, upgrade DB2 Connect PE to the appropriate Fix Pack.

For information regarding the supported Fix Pack version, see *System Requirements and Supported Platforms* on Siebel SupportWeb.

If you previously installed a DB2 Connect run-time client, the DB2 Connect installer upgrade adds only the functionality required for the existing client. (This is also the case if you have a DB2 server or SDK installed on your workstation.)
- 3** On the workstation machine, catalog your OS/390 databases as appropriate, using the DB2 Client Configuration Assistant or the DB2 Command Line Processor.

After installation, use a standard DB2 UDB client to access DB2 Connect.

Additional Postinstallation Steps for DB2 Connect

This section assumes that you have installed DB2 Connect using the installation defaults and have applied the necessary IBM Fix Pack. It also assumes the network connectivity (TCP/IP and FTP) has been set up from the DB2 Connect server to the main frame.

In order to use DB2 Connect effectively with Siebel applications, you will perform some additional steps (defining an alias and BINDing) and some additional configuration. DB2 Connect has a series of configuration parameters that define pooling and thread reuse, among other things. These configuration changes vary depending upon site-specific details, and whether the Enterprise or Personal version of DB2 Connect was installed. The details are provided to assist with this configuration.

Changes to these parameters change the database manager configuration file. For more information refer to your IBM documentation for the details on how to change the database manager configuration parameters.

Defining a Database Alias and Testing a Connection

The information in this section describes how to define a database alias and test a connection.

In creating a connection to DB2 on the zSeries, a connection is defined with a database alias that has the proper connection information for your database. This alias is the connect string that you specify when you install the Siebel Server. Record the connect string in the copy you made of [Appendix A, "Deployment Planning Worksheet."](#)

To define a database alias

- To define your database alias, use the DB2 Configuration Assistant or the DB2 Command Line Processor.

NOTE: If you define database aliases, you must add the `TXN ISOLATION` parameter to the database alias entry in the `db2cli.ini` file, and set it to have a value of 1.

For more information, refer to the IBM documentation.

To test the connection on the Windows platform

- 1 Launch the Configuration Assistant, highlight the database alias that you just created, and right click.

This brings up an additional dialog that allows you to test the connection.

- 2 Test the connection on the Windows platform using one of the methods below:

Supply the userid and password and choose both ODBC and CLI as the test parameters.

- If using the DB2 command window, enter
DB2 connect to *dbalias* user *database userid* using *database password*
- If using the DB2 Command Line Processor, enter
connect to *dbalias* user *database userid* using *database password*

If the connection is successful you will see output similar to the following:

```
Database Connection Information
Database server          = DB2 OS/390 7.1.1'1
SQL authorization ID     = SADMIN
Local database alias     = Q105
```

To test the connection on the UNIX platform

- 1 Open a UNIX shell.
- 2 Source the `siebenv.csh` file.
- 3 Enter the following command:

DB2 connect to *database_alias* user *user_ID* using *password*

where:

user_ID = a valid user name on the DB2 host.

password = the appropriate password for that user name.

If your connection is valid, database connection information appears; for example:

```
Database Connection Information
```

Database server = DB2 OS/390

SQL authorization ID = SADMIN

Local database alias = PS01

To verify the connection to the database using the Configuration Assistant

- 1** Launch the IBM DB2 Configuration Assistant.
- 2** Choose the appropriate database alias and from the Selected option section choose Test Connection.
- 3** From here you may test the CLI and ODBC connections using an authorized userid and password.

The following message indicates a successful connection:

CLI connection tested successfully.

ODBC connection tested successfully.

Binding the DB2 Connect Packages

DB2 Connect is installed with .bnd files that bind to the host server. You can bind these packages using the default collection ID, or you can create a new collection ID for these packages. See the discussion on security. The BIND command is only issued once per subsystem, and version/IBM Fix Pack.

The Bind options are as listed:

- Use DB2 Configuration Assistant's BIND option to bind the packages. Highlight the dbalias, right click, and choose the BIND option from the pop-up dialog.
- Issue explicit BIND commands. Refer to IBM documentation for further details on the BIND command.

The BIND command is issued at the time of installation and when subsequent IBM Fix Packs are applied. As indicated earlier, you will BIND once per subsystem. Failure to BIND will result in SQL errors or authorization errors when a client without BINDADD authority attempts to connect and the DB2 software attempts to perform an autobind. Please refer to the IBM documentation for more information on the BIND command.

Configuring DB2 Connect for DB2 Connect Enterprise Edition Only

The DB2 Connect EE is a server version that allows multiple users to connect to DB2 on the DB2 host. Because it is a server, you must complete additional configuration steps:

- You must define the maximum number of concurrent users (MAXAGENTS).

- You must specify whether or not to enable Connection Concentration.

Connection Concentration provides additional workload balancing opportunities for DB2 Data Sharing by allowing connections to be rebalanced across DB2 Data Sharing members at the commit phase instead of just at the time of initial connection. This dynamic rebalancing of connections across members can be beneficial after a planned or unplanned outage to DB2.

NOTE: Connection Concentration is disabled by default during the installation of DB2 Connect.

Connection Concentration Configuration Parameters

As installed, the default settings for the configuration parameters that control connection concentration are:

- Priority of agents (AGENTPRI) = SYSTEM
- Maximum number of existing agents (MAXAGENTS) = 100. MAXAGENTS is the maximum number of worker agents. This value represents the maximum number of concurrent connections to DB2 for z/OS.
- Agent pool size (NUM_POOLAGENTS) = 50 (calculated). NUM_POOLAGENTS is the maximum number of idle pool agents. This value represents the number of concurrent connections to DB2 for z/OS. Setting this parameter to 0 disables connection pooling.
- Initial number of agents in pool (NUM_INITAGENTS) = 0
- Maximum number of coordinating agents (MAX_COORDAGENTS) = (MAXAGENTS - NUM_INITAGENTS)
- Maximum number of concurrent coordinating agents (MAXCAGENTS) = MAX_COORDAGENTS
- Maximum number of client connections (MAX_CONNECTIONS) = MAX_COORDAGENTS

NOTE: When MAX_CONNECTIONS = MAX_COORDAGENTS, Connection Concentration is not enabled.

If you do not want to enable Connection Concentration and if you have not made any adjustments to the parameters, then the only configuration to do is to set the number of MAXAGENTS to the number of concurrent users that you expect on the system. To set this number, issue an update dbm cfg using statement. For more information on updating the database manager configuration, see the vendor documentation on the IBM Web site.

If you are using Siebel Connection pooling (and it is recommended that you do), the value you choose for MAXAGENTS determines the value you specify for the MaxSharedDbConns value. Further, it determines the value you specify for MAXDBATS on the z/OS. For more information about this feature, see [“Database Connection Pooling” on page 164](#).

To Enable Connection Concentration

- Set the value of the maximum number of client connections so that MAX_CONNECTIONS is greater than MAX_COORDAGENTS (MAX_CONNECTIONS = MAX_COORDAGENTS + 1).

Setting MAX_CONNECTIONS = MAX_COORDAGENTS + 1 enables the Connection Concentration dynamic rebalancing capabilities and still allows the transaction pooling to be handled by Siebel pooling.

NOTE: If you enable Connection Concentration, you must first set the value for the maximum number of agents (MAXAGENTS) parameter; this value determines the value of other parameters. Set the value of the MAXAGENTS parameter as appropriate for your site; this value varies according to the memory available and the number of server connections expected.

Setting Up the Subsystem

Siebel Systems supports production deployments of Siebel eBusiness Applications in both separate and shared DB2 subsystems. However, setting up a separate subsystem is preferable for Siebel applications, particularly for larger deployments, for the following reasons:

- **DSNZPARM optimization.** The DSNZPARMs used with this release of Siebel eBusiness Applications are optimized for the Siebel application, but might not be optimal for use with non-Siebel applications. It is recommended, but not required, that you run the Siebel eBusiness Application on its own subsystem.
- **OLTP and OLAP characteristics.** Siebel applications possess the characteristics of both an OLTP and OLAP application. However, most other vendors' applications have the characteristics of either one or the other.
- **DB2 maintenance requirements.** This release of Siebel eBusiness Applications requires the application of a number of maintenance activities that are documented in an informational APAR that can be located on IBM's Web site.

CAUTION: It is crucial that you apply all maintenance releases in the informational APAR to the subsystem that you set up. Otherwise, operating anomalies and unpredictable behavior might occur.

- **System catalog locking during Siebel database recovery.** With the Siebel 7.7 approach of defining one table space per a database, this is not a problem. However, if you use database layouts with multiple table spaces per database, recovery of the Siebel Schema might affect other applications on a shared DB2 subsystem if the system catalog becomes locked during the recovery process. Because recovery typically requires the restoration of all Siebel table spaces, locking could last many hours.

Even though a shared DB2 subsystem is preferable for implementing Siebel eBusiness Applications, there are numerous successful Siebel deployments in shared DB2 subsystems.

About Unicode Conversion Tables

Siebel eBusiness Applications, Version 7.5 introduced Unicode character set support. However, Siebel Systems does not support Unicode on DB2 for z/OS. As a result, if the Siebel Unicode-enabled application server is to communicate properly with the DB2 for z/OS database, you must install Unicode Conversion Services. Conversion errors can result if the z/Series Unicode Conversion Services images do not include the required conversion tables.

You must configure the z/Series Unicode Conversion Services with the conversions required by the codepages specific to your DB2 subsystem. Because of the difficulty in determining all the conversions required for the conversion image, it is recommended that you use the prebuilt image provided on z/OS V1.R2 and later, as provided by the IBM APAR OA04069. This preloads a Unicode conversion image with many tables which support a substantial number of codepage conversions.

Choosing the Database CCSID

Siebel Systems supports both ASCII- and EBCDIC-based coded character set IDs (CCSIDs) on DB2 UDB for z/OS and OS/390. The sort order used in list applets, among others, depends on the database CCSID.

Review the *Release Notes* on Siebel SupportWeb for information about known restrictions; for example, the following features are not supported on databases with EBCDIC code pages:

- Web Client migration
- Siebel Data Warehouse
- Siebel Dun and Bradstreet server components

You should run development databases with ASCII code pages, because databases with EBCDIC code pages do not support the following procedures in a development environment upgrade:

- Merging prior configuration changes into a new custom configuration repository (this applies to upgrades only)
- Compiling a new Siebel repository file (.srf file) from the new repository

You can partition table spaces on a database with either an ASCII or an EBCDIC code page, so that limit keys reflect the sort sequence difference between ASCII and EBCDIC. For more information, see [Chapter 5, "Configuring the Database Layout."](#)

You can deploy either an ASCII- or an EBCDIC-based CCSID to create your Siebel Schema, but it is generally recommended that you use ASCII.

- Use an ASCII-based encoding scheme to reduce the overhead required for character conversion. Character conversion is performed by DRDA, between the database and the midtier ASCII servers.
- Use an EBCDIC-based encoding scheme if you want to join Siebel tables with other EBCDIC DB2 tables.

CAUTION: DB2 version 7 only supports a single code page for tables referenced within a single SQL statement.

Setting the CCSID

The CCSIDs must be configured in the fields ASCII CODED CHAR SET or EBCDIC CODED CHAR SET on the DB2 installation panel DSNTIPF. The field DEF ENCODING SCHEME on the same panel determines whether the default encoding scheme is ASCII or EBCDIC.

NOTE: The field MIXED DATA on DSNTIPF should equal NO for all supported languages except Japanese.

Using the Data Distribution Facility and Workload Manager

The Data Distribution Facility (DDF) is a component of DB2 that provides a connection between your DB2 subsystem and remote clients and systems. Specifically, it communicates with DB2 Connect for remote application access. You must configure and activate DDF before you install Siebel eBusiness Applications. DDF setup is described in the vendor documentation on the IBM Web site.

Workload Manager (WLM) is an integrated component of z/OS that manages system resources across z/Series subsystems. You must configure WLM for use with DDF threads and Siebel application stored procedures and user-defined functions (UDFs).

In relation to DB2, WLM manages incoming requests for DB2 stored procedures and UDFs, and allocates resources to process them. If you do not have WLM installed and configured, you cannot use Siebel components that require stored procedures and UDFs. Stored procedures are used with EIM (when UPDATE STATISTICS=TRUE) and the Siebel Upgrade application component. UDFs are used with EIM Export and currency aggregations. For information on setting up WLM, see the vendor documentation on the IBM Web site.

DSNZPARM Parameters for Setting Up the Subsystem

For optimum operation of Siebel eBusiness Applications, use the recommended settings in [Table 7](#) to configure the parameters within DSNZPARM.

The following parameter values are required by Siebel eBusiness Applications:

```
CACHEDYN=YES
CDSSRDEF=1
EXTSEC=YES
MXTBJOIN=128
PARTKEYU=YES
RETVLCFK=NO
OPTCCOS1=ON
OPTSUBQ1=YES
```

NOTE: PARTKEYU is only required if partitioning is used with the Siebel Schema.

From the Siebel Systems standpoint, there are three categories of DSNZPARMs: required, recommended, and the remaining DSNZPARMs. Generally it is recommended that the default values for the DSNZPARMs, which are not explicitly listed here, could be modified by your DBA if it benefits your environment. Modifications of such DSNZPARM values should be closely monitored by your DBA to assess the impact on your DB2 subsystem.

You can configure some parameters online, but other parameters require you to shut down DB2 to configure them. For information on which parameters can be updated online, refer to the vendor documentation on the IBM Web site.

Table 7. Database Manager Configuration Parameters (DSNZPARM)

Parameter	Explanation	Recommended Setting
DSN6SPRM		
CACHEDYN	Turns on dynamic statement caching. This is a required value for Siebel eBusiness Applications.	YES (required)
CDSSRDEF	Turns off parallelism for dynamic statements. This is a required value for Siebel eBusiness Applications.	1 (required)
CONSTOR	Compresses storage on a regular basis. Set this value to YES if DBM1 storage is an issue.	YES
EVALUNC	Allows predicate evaluation on uncommitted data. This is a recommended Siebel eBusiness Applications value.	YES
IRLMRWT	Time in seconds before a time-out occurs. This is a recommended Siebel eBusiness Applications value.	300
MINSTOR	Compresses storage on a regular basis. Set this value to YES if DBM1 storage is an issue.	YES
MXTBJOIN	Sets the maximum number of tables in a join. The default (15) is not sufficient for Siebel eBusiness Applications. NOTE: Unlike other DSNZPARMs that cannot be updated through the installation panels, this parameter must be updated manually.	128 (required)
NPGTHRSH	Allows small tables to use indexes. This is a recommended Siebel eBusiness Applications value.	10
NUMLKUS	Number of locks per user. This value should be monitored and set by the DBA. If you experience persistent locking problems, consider setting the parameter to 0 (unlimited number of locks).	100,000
OPTCCOS1	Allows multiple predicate optimizer enhancements.	ON (required)
OPTSUBQ1	Allows non-correlated subquery optimizer enhancements.	YES (required)

Table 7. Database Manager Configuration Parameters (DSNZPARM)

Parameter	Explanation	Recommended Setting
PARTKEYU	Allows update of partitioning keys. This is a required Siebel eBusiness Applications value.	YES (required)
RETVLCFK	Allows index-only access of varying-length characters. This value should be set to NO due to padding. This is a recommended Siebel eBusiness Applications value.	NO (required)
DSN6SYSP		
CHKFREQ	Avoids frequent checkpoints in a high-update environment. DBAs should monitor and set this value for between 10 and 20 minutes.	500, 000 (or 15 when FREQUENCY TYPE = MINUTES)
CONDBAT	Maximum number of concurrent remote connections. The DBA should monitor and set this value.	10, 000
EXTSEC	Allows DB2 Connect to receive more complete error messages. Allows the changing of passwords from DB2 Connect. This is a required Siebel eBusiness Applications value.	YES (required)
MAXDBAT	Maximum number of database threads (DBAT). The DBA should monitor and set this value.	500
DSN6FAC		
CMTSTAT	Allows a greater number of remote threads without affecting storage. Enables DDF thread pooling. This is a recommended Siebel eBusiness Applications value.	INACTIVE
IDTHTOIN	Number of seconds before an idle thread is canceled. Prevents long-running threads from holding resources. The DBA should monitor and set this value.	600
POOLINAC	Number of seconds an inactive thread remains in the DDF pool. The DBA should monitor and set this value.	120

NUMLKUS Considerations

If a *resource unavailable* error occurs, because NUMLKUS has been exceeded while performing a Siebel operation, do one of the following:

- Increase the NUMKLUS parameter value.
- Issue COMMIT statements more frequently, and reduce the number of update and insert operations while using Siebel eBusiness Applications.

NOTE: The NUMKLUS parameter value is important when running large EIM batches and during the use of Siebel Remote for the initial database extract. If this value is too small, EIM runs or the database extract can fail.

Should EIM fail due to NUMLKUS being exceeded, do one of the following:

- Reduce the size of the batch.
- Increase the value of NUMLKUS.

IDTHTOIN Considerations

The IDTHTOIN DSNZPARM recommended setting is 600. (The parameter is specified in seconds.) This value must be monitored in your installation to be sure that it is set properly. The IDTHTOIN will time out active threads that have been idle (no network traffic) for more than 600 seconds. (This is the recommended value; your value may differ.)

The events that may occur when an active thread is timed out include:

- The thread is *not* returned to the DB2 Connect thread pool for re-use.
- When the client or component tries to re-use that DB2 Connect thread, the thread will not be available. The client or component must be able to issue a full connect in order to continue processing the work.
- If the component or client is not capable of issuing a full connect, the client will receive a SQL30081N SQLCODE. An example of an error message you may receive if the component or client cannot reconnect are:

```
[IBM][CLI Driver][DB2] SQL30081N  A communication error has been detected.  
Communication protocol being used: "TCP/IP". Communication API being used:  
"SOCKETS". Location where the error was detected: "". Communication function  
detecting the error: "recv". Protocol specific error code(s): "*", "*", "0".  
SQLSTATE=08001
```

Another consideration for specifying the IDTHTOIN is that connection pooling on the ODBC driver must be turned off. Using connection pooling with the driver is not certified with Siebel 7.

Estimating the Storage Space Required

The space needed by DB2 varies primarily based on the total number and types of users supported. Consult the IBM DB2 UDB for z/OS and OS/390 technical documentation for more information on these requirements.

The space required for Siebel data and indexes varies depending on what Siebel functionality you implement and the amount and nature of data supporting it. The process for making accurate database size calculations is a complex one involving many variables. The following guidelines will assist you in the process:

- Determine the total number, and types, of users of Siebel eBusiness Applications (for example, 500 sales representatives and 75 sales managers).
- Determine the Siebel functionality that you will implement and the entities required to support them. Typically, the largest entities are:
 - Accounts
 - Activities
 - Contacts

- Forecasts
- Opportunities
- Service Requests
- Estimate the average number of entities per user (for example, 100 accounts per sales representative) and calculate an estimated total number of records per entity for your total user base.
- Calculate the average record size per entity and multiply by the total number of records using standard sizing procedures for your specific database and the Siebel Schema definition. Typically, these entities span multiple physical tables, all of which must be included in the row size calculation. This calculation determines the estimated data sizes for the largest entities.
- Add extra space for the storage of other Siebel data. A rough guideline for this amount is one-half the storage required.
- Allow for a margin of error in your total size calculation.
- Factor growth rates into your total size calculation.

Allocating Space for Buffer Pools and Storage Groups

The following example illustrates how to alter space for buffer pools, in preparation for installing the Siebel Schema, using a group ID.

```
ALTER BUFFERPOOL (BP32K1) VPSIZE (4000)
GRANT USE OF BUFFERPOOL BP32K1 TO PUBLIC;
```

The following example illustrates how to allocate space and access to storage groups using a group ID.

```
CREATE STOGROUP SIEBEL VOLUMES ('*') VCAT SIEBEL;
GRANT USE OF STOGROUP SIEBEL TO PUBLIC;
```

Planning the Distribution of Database Objects

Be sure that you properly distribute the following types of database objects:

- Database log or archive files
- Temporary work space used by the database
- Tables and indexes
- Partitions
- CHAR large objects (CLOBs)

In most implementations, the Siebel tables listed in [Table 8](#) and their corresponding indexes are either the most commonly used tables or the largest tables. For example, the tables S_EVT_ACT, S_PARTY, and S_ORG_EXT are large in all enterprise-level deployments of Siebel eBusiness Applications. These tables and their indexes should be separated across devices. As a general rule, indexes should be on different physical devices from the tables for which they are created.

Table 8. Most Frequently Used and Largest Siebel Tables

Table Names			
S_ACCNT_POSTN	S_CONTACT_FNX	S_FN_ACCNT2_FNX	S_ORG_EXT_X
S_ACT_CONTACT	S_CONTACT_FNXM	S_INVC_ITM_DTL	S_ORG_EXT_XM
S_ACT_EMP	S_CONTACT_X	S_LIST_COL_INTL	S_ORG_TERR
S_ADDR_ORG	S_DOCK_TXN_LOG	S_NOTE_ACCNT	S_PARTY
S_ADDR_PER	S_DOCK_TXN_SET	S_NOTE_DEFECT	S_PARTY_PER
S_ASSET	S_ETL_R_IMAGE	S_NOTE_OPTY	S_POSTN_CON
S_ASSET_CON	S_ETL_VR_IMAGE	S_OPTY_ORG	S_PRSP_CONTACT
S_ASSET_POSTN	S_EVT_ACT	S_OPTY_ORG_CMPT	S_QUOTE_ITEM
S_ASSET_REL	S_EVT_ACT_FNX	S_OPTY_POSTN	S_SLSCRDT_VAL
S_CAMP_CON	S_EVT_ACT_X	S_OPTY_TERR	S_SRV_ACT
S_COMMUNICATION	S_EVT_MAIL	S_ORG_BU	S_TMSHT
S_CON_ADDR	S_EXP_ITEM	S_ORG_EXT	S_TMSHT_APPR
S_CONTACT	S_EXP_RPT_APPR	S_ORG_EXT_FNX	S_TMSHT_ITEM
S_CONTACT_BU	S_FN_ACCNT1_FNX	S_ORG_EXT_FNXM	S_TMSHT_LN

Estimating the Number of Database Objects You Need

Siebel Systems ships over 3,000 tables, table spaces and databases, and over 15,000 indexes for Siebel eBusiness Applications and over 4,000 tables and 21,000 indexes for Siebel Industry Applications.

The number of objects created in the Siebel Schema depends on which Siebel product line you purchase. This consequently determines how much space you must allocate. It is up to you to determine which of the objects shipped might actually be required for your deployment, based on your business needs.

CAUTION: You must not drop unused tables; doing so causes Siebel Systems to discontinue technical support of your deployment. If you need to drop unused, non-unique Siebel indexes on the z/Series, contact Siebel Technical Support for assistance.

Siebel eBusiness Applications ships all table space and index schema definitions with the `DEFINE NO` option as the default. The DB2 `DEFINE` option specifies if the underlying data sets for a table space or index are created physically. Because the Siebel default is `DEFINE NO`, this means that no physical data sets are created until data is actually inserted into the table space.

NOTE: `DEFINE NO` is only applicable to DB2-managed (`STOGROUP`) data sets and is ignored for user-managed (`VCAT`) data sets and for LOB table spaces.

5

Configuring the Database Layout

This chapter describes the logic behind the storage control file templates that are shipped with Siebel eBusiness Applications for database schema configuration during the installation process. The primary audience for this chapter is the DB2 DBA.

This chapter consists of the following topics:

- [“About the Siebel Database Server File Installation”](#)
- [“About the Siebel Storage Control File Templates” on page 44](#)
- [“About Siebel Objects” on page 45](#)
- [“About Creating or Modifying the Database Layout” on page 52](#)
- [“About the Siebel Database Storage Configurator” on page 53](#)

About the Siebel Database Server File Installation

The Siebel Database Server file installation process uses two types of Siebel control files:

- **DDL control file.** The DDL control file contains the logical definitions for all Siebel tables and indexes for the current version of the Siebel eBusiness Application. The DDL control file (ddl.ctf or schema.ddl) is read-only to protect the integrity of the Siebel data model.
- **Storage control file.** The storage control file contains the physical database layout. The storage control file is specific to DB2 UDB for z/OS and OS/390.

Carefully review the storage control file (or the resulting schema.sql file that is based on the storage control file) and adjust the database layout to satisfy your database requirements. You can modify any option available for each database object, for example, by fine-tuning FREEPAGE and PCTFREE for a particular table space or a group of table spaces.

CAUTION: It is especially important that you review SEGSIZE, PRIQTY, and SECQTY for the table spaces that will contain the major tables for your implementation. Siebel Systems has appropriately sized the table spaces used for Siebel repository tables, but because sizing requirements for base Siebel tables vary from deployment to deployment, it is your responsibility to set storage parameters capable of accommodating your installation.

Together, the DDL control file and the storage control file provide Siebel eBusiness Applications with all the information necessary to create DDL for the Siebel schema in an output file (by default named schema.sql).

There are two methods you can use to execute the SQL files from a designated Siebel Server:

- **Method 1.** Automatic execution from Siebel Server using an ODBC utility (such as `odbcsql` from Siebel Systems) or a version control tool for file execution. To use this method, review [“Configuration Options for DB2 Connect” on page 26](#) to determine the configuration appropriate to the Client your enterprise supports, then follow the instructions in [“Installing Required IBM Fix Packs” on page 27](#).
- **Method 2.** Direct execution on DB2 host using Siebel-provided scripts, your own FTP, or any other file transfer program to transfer files to DB2 host, then use a native IBM utility such as `SPUFI` for file execution. This method is outside the scope of this guide. If you want to execute SQL using mainframe native tools, refer to IBM documentation.

About the Siebel Storage Control File Templates

The storage control file is unique to the DB2 UDB for z/OS and OS/390 platform for Siebel eBusiness Applications. The storage control file contains storage information (for example, partitioning indexes, table spaces, and storage groups) that is used as the basis for the storage layout of your Siebel database. Even if you are using a preconfigured storage layout, make sure that the layout is valid for your Siebel schema.

Starting with release 7.7, the Siebel Schema layout consists of one table per table space and one table space per database. This layout reduces logging, removes concurrency issues, and allows for more databases to be used to achieve high database parallelism.

The Siebel application installation process installs three storage control file templates in the `DBSRVR_ROOT\DB2390` directory for Windows and the `$DBSRVR_ROOT/db2390` directory for UNIX. Select one of the templates that follow based on your partitioning needs and your encoding scheme (if you are using partitioning):

- **storage_np.ctl.** This template contains database storage layout for Siebel Schema with one table per table space, and one table space per database. No partitioned table spaces are provided in this storage control file.
- **storage_p.ctl.** This template is the database storage layout for Siebel Schema with partitioning for ASCII encoding. Every nonpartitioned table resides in its own segmented table space. Each table space resides in its own database.
- **storage_p_e.ctl.** This template is the database storage layout for Siebel Schema with partitioning for EBCDIC encoding. Every nonpartitioned table resides in its own segmented table space. Each table space resides in its own database.

The layouts of the storage control file templates are generic, and they require customization by your DBA to suit the needs of your enterprise, for example, to change the default space allocation for Siebel objects, seed data, and views.

While Siebel-provided templates might be sufficient for your development environment without any changes, your DBA needs to review and modify the database storage layout to best fit the needs of your deployment in a production environment.

The storage control file templates are the starting point for your customization process. Before you select a Siebel storage control file template, you should understand the logic behind how objects are grouped, and you should understand the object naming conventions. Knowledge of these concepts will help you when you perform customizations and will help you understand how your operations affect the underlying data sets.

About Siebel Objects

An understanding of Siebel objects and object naming conventions is a foundation for understanding how your customizations affect the underlying data sets. The following Siebel objects are described in this section:

- [“About Substituting Values Using Symbolic Variables” on page 45](#)
- [“About Default Objects in Storage Control Files” on page 46](#)
- [“About Buffer Pools Used in Storage Control File Templates” on page 47](#)
- [“About Using Storage Groups” on page 47](#)
- [“About Databases in Storage Control Files” on page 48](#)
- [“About Tables in Storage Control Files” on page 48](#)
- [“About Table Spaces in Storage Control Files” on page 49](#)

For an explanation of object naming conventions, see [“About Object Naming Conventions” on page 50](#).

About Substituting Values Using Symbolic Variables

The storage control file contains objects (such as databases, table spaces, and tables) that define the physical storage layout of your database. Many of the options in the objects are the equivalent of options in DB2 SQL statements.

While most options within the objects are defined by actual values (for example, SegSize=32), some options include symbolic variables that will be substituted by the values in [Table 9](#). Review [Table 9](#) for a list of symbolic variables from the storage control file templates that will be substituted with the actual values that comply with your organization's standards.

Table 9. Symbolic Variables

The symbolic variable...	Will be substituted with the actual value for...
\$DbnamePrefix	Database Name Prefix
\$StogroupTables	Table Storage Group for Tables
\$StogroupIndexes	Index Storage for Indexes
\$IndexBufferPool	Index Buffer Pool Name
\$4KBufferPool	4-KB Buffer Pool Name
\$16KBufferPool	16-KB Buffer Pool Name
\$32KBufferPool	32-KB Buffer Pool Name
\$DbType	Encoding Scheme (acceptable values are either ASCII or EBCDIC)

About Default Objects in Storage Control Files

The first object in the storage control file is a Defaults object. If no actual value exists for a given object, but a value for that object is required by a DB2 statement, then the default value is either derived from the Defaults object or inherited from a higher-level object.

The following example illustrates what the Defaults object looks like in the storage control file.

```
[Object 1]
Type      = Defaults
Name      = Defaults
Database  = S1DB0000
Tablespace = S1EBTS00
Stogroup   = SYSDEFLT
IndexStogroup = SYSDEFLT
Bufferpool = BP1
IndexBp    = BP2
SegSize    = 32
LockSize   = Page
LockMax    = 0
PriQty     = 48
SecQty     = 1440
Defi ne    = No
PctFree    = 30
FreePage   = 4
Erase      = No
CCSID      = ASCII
```

Compress = No
Type = Default s

You can override default values for any given object by explicitly defining the value in the object. For example, if the Tablespace object does not contain PriQty or SecQty values, then these values are inherited from the PriQty and SecQty values defined in the Defaults object. However, if the values are defined in the Tablespace object, then the explicitly provided value overrides the default and is used in the output DDL.

NOTE: If you create an extension table in Siebel Tools, and you use the apply process without creating new underlying objects for this table in the storage control file, then the necessary storage layout is derived from the default definitions. For information about the apply process, see [“Applying Schema Changes to the Target Database” on page 155](#).

About Buffer Pools Used in Storage Control File Templates

Siebel eBusiness Applications use four buffer pools in the storage control file templates:

- One 4-KB buffer pool for 4-KB table spaces (the default name is BP1)
- One 4-KB buffer pool for all indexes (the default name is BP2)
- One 16-KB buffer pool for 16-KB table spaces (the default name is BP16K1)
- One 32-KB buffer pool for 32-KB table spaces (the default name is BP32K1)

Multiple buffer pools can be used for each page size. If you require multiple buffer pools, use the Siebel Database Storage Configurator (dbconf. xls) or any other method described in [“Preparing the Database Layout” on page 101](#) to enter these values.

Your DBA needs to activate and grant use of buffer pools before you can perform the Siebel installation. See [“Allocating Space for Buffer Pools and Storage Groups” on page 40](#) for an example of buffer pool definitions.

About Using Storage Groups

Siebel eBusiness Applications require one storage group for indexes and one storage group for tables (data). Before performing your installation, ask your DBA to provide you with one storage group name for each. You can use the same name for data and indexes, or you can use different names. You can set up storage group usage in any way that makes sense for your deployment.

See [“Allocating Space for Buffer Pools and Storage Groups” on page 40](#) for an example of storage group definitions.

About Databases in Storage Control Files

In the Siebel eBusiness Applications database storage layout, each table resides in its own table space, and each table space resides in its own database. The number of databases you receive depends on which modules your organization purchased. You can regroup table spaces in databases, as required for your enterprise.

NOTE: The maximum number of databases is 64,000 per one DB2 subsystem.

The following example illustrates what a Database object looks like in the storage control file.

```
[Object 4]
Type       = Database
Name       = S1DB2532
LockSize   = Page
```

The Database object illustrated above translates into the following DDL output:

```
CREATE DATABASE S1DB0001 CCSID ASCII BUFFERPOOL BP1 /
```

In Siebel eBusiness Applications, CCSID is defined on the database level only. CCSID is always taken from the Defaults object.

There is no LockSize used in the `CREATE DATABASE` statement, because LockSize is a default value for all table spaces that belong to a given database.

The names of databases in storage control files are comprised of two parts. The first part of the name is a variable, *\$DbnamePrefix*. The second part of the name is a four-digit number. This four-digit number in the database name is the same four-digit number used in the table space name that resides in this database. You can substitute the database name prefix with a literal value up to four characters long (for example, S1DB) that conforms to your organization's naming convention. You can change the value of the database name prefix either during the installation process or by changing it in the storage control file. For more information on the table space name structure, see [“About Table Space Naming Conventions” on page 50](#).

About Tables in Storage Control Files

The following example illustrates what the Table object looks like in the storage control file.

```
[Object 7149]
Type       = Table
Name       = S_ORG_EXT
Database   = S1DB1465
Tablespace = H1465000
Clob       = No
```

The Table object illustrated above translates into the following DDL output:

```
CREATE TABLE SIEBTO.S_ORG_EXT (
  ROW_ID VARCHAR(15) NOT NULL,
  CREATED_TIMESTAMP DEFAULT NOT NULL,
  CREATED_BY VARCHAR(15) NOT NULL,
  ...
```



```
VAT_REGN_NUM VARCHAR(30),
DIRECTIONS LONG VARCHAR) IN SIEBTO.H1465000 /
```

Siebel eBusiness Applications include one template per encoding scheme when partitioning is used and one template per encoding scheme for non-partitioning.

Most tables use page-level locking by default, but a few exceptions use row-level locking. [Table 10](#) lists tables that use row-level locking.

Table 10. Tables with Row - Level Locking

Row-Level Locking
S_ESCL_ACTN_REQ
S_ESCL_LOG
S_ESCL_REQ
S_DOCK_TXN_LOG
S_DOCK_TXN_LOGT
S_DOCK_TXN_SET
S_DOCK_TXN_SETT

NOTE: LockSize is not defined on the table level, but in the corresponding table space.

[Table 11](#) lists tables that use table-level locking.

Table 11. Tables with Table - Level Locking

Table-Level Locking
S_DOCK_INIT_ITEM
S_DOCK_INITM_XX

About Table Spaces in Storage Control Files

Siebel tables are created in one of three table space sizes, 4 KB, 16 KB, and 32 KB. Most tables are defined within a 4-KB table space.

NOTE: Users can create new table spaces, and you can regroup tables as required for your enterprise.

The following example illustrates what the segmented Tablespace object for the Siebel repository looks like in the storage control file.

```
[Object 3387]
Type       = Tablespace
Name       = H1004000
```

```

Database    = SI DB1004
LockSi ze   = Page
Bufferpool  = BP1
Defi ne     = No
Parti ti ons = 0

```

This Tablespace object translates into the following DDL output:

```

CREATE TABLESPACE H1004000 IN SI DB1004
USING STOGROUP SYSDEFLT
PRI QTY    48 SECQTY 1440 FREEPAGE 4 PCTFREE 30
DEFINE NO
SEGSI ZE 32
BUFFERPOOL BP1 LOCKSI ZE PAGE LOCKMAX 0 COMPRESS NO /

```

NOTE: If you compare the input and the output, you can see that there are more options in the DDL output. This is because some values, such as `FREEPAGE` and `PCTFREE`, are taken from the default values.

Object numbering can change when you perform a storage control file extract. Therefore, object numbering can be different from one storage control file to another (except for default storage objects, for which numbering remains consistent).

About Object Naming Conventions

When you work with Siebel objects you must understand Siebel object naming conventions. An understanding of object naming conventions allows you to keep track of existing Siebel objects and to successfully create and maintain your own objects.

If you understand Siebel naming conventions, you can more easily identify underlying datasets (VSAM files); for example, the name of the underlying datasets for H0401000 table space might look like Q10E.DSNDBC.SIEBTO.H0401000.I0001.A001 where the fourth node represents the table space name.

NOTE: Siebel utilities are able to recognize and use custom objects only if they follow Siebel naming conventions; therefore, it is strongly recommended that you use the Siebel naming conventions for objects that you create.

About Table Space Naming Conventions

The table space naming convention applies to all Siebel tables, whether you created the tables or the tables were shipped as part of your Siebel eBusiness Application.

Table spaces are named with eight characters (for example, H0401000) that consist of three elements in the following order:

- 1 A leading letter (H, V, or S) designates the product group that the table belongs to.
 - H represents Horizontal (also known as Core).
 - V represents Vertical (also known as Siebel Industry Applications or SIA).

- S represents your custom tables and auxiliary indexes (those that you create for the Siebel application).
- 2 The four-digit number that was assigned to the base table and is stored in GROUP_CD column of S_TABLE table in Siebel Repository, for example, 0401. (This number cascades down to table spaces, auxiliary tables, and auxiliary indexes.)

Table names are stored in the repository table S_TABLE, and the table number is stored in the Group Code column (GROUP_CD) in S_TABLE. Table and Group names can be viewed in Siebel Tools under Table > Object.

NOTE: The *Group* property for tables is editable for those tables created by you. You should not edit *Group* for the tables provided in the Siebel application.

- 3 The ending three-digit number is always 000 for table spaces on the database storage layout.

CAUTION: When you create new custom tables and auxiliary indexes, use the prefix S, follow it with a unique four-digit number greater than 6000, then end with the three-zero suffix (000), for example, S6001000. This maintains object naming conventions, and prevents custom objects from being created with random names.

About Naming Conventions for Auxiliary Object

Names for auxiliary objects such as character large objects (CLOBs) must follow a naming convention that provides space for the auxiliary table, table space, and index for each partition of a partitioned CLOB.

Auxiliary objects are named with eight characters (for example, A0401000) that consist of three elements in the following order:

- 1 A leading letter designates the product group that the table belongs to. Use the following matrix to determine which prefix to use.

	Tables	Table Spaces	Table Indexes
Horizontal	X	A	I
SIA and Custom ("S")	Y	B	J
Default Group Names Assigned to Objects with No Leading "S"	Z	C	K

- 2 A four-digit number that uniquely identifies the table space, for example, 0401.

NOTE: Default numbering of auxiliary objects starts from 0001.

- 3 The ending three-digit number must be all zeros: 000. (Every name ends in 000.)

About Partitioning Triggers

Partitioning Triggers are named with eight characters (for example, PTH0401) that consist of three elements (in the following order):

- 1 The leading two letters must be PT to designate that this is a partitioning trigger.

- 2 The third letter (H, V, or S) designates the product group or component that the trigger belongs to.
 - H represents Horizontal (also known as Core).
 - V represents Vertical (also known as Siebel Industry Applications or SIA).
 - S represents partitioning triggers used for Assignment Manager or other components.
- 3 The ending four-digit number uniquely identifies the trigger based on the table number, for example, 0401.

The trigger name is normally related to a corresponding table space name; for example, the trigger name PTH0401 relates to the table space name H0401000.

About Creating or Modifying the Database Layout

There are two methods by which you can modify the database storage layout for your deployment:

- **Method 1. Adjust the Storage Control File to Reflect Your Database Layout.** Adjust the object definitions and options in the storage control file to reflect your database storage layout. When you subsequently perform a customized database installation ([“Performing a Custom Installation” on page 114](#)) or when you use database utilities, select your new storage control file when you are prompted for the storage control file input parameter. The resulting output file, schema.sql, contains the DDL that reflects your modifications.

If this method is used, it will be necessary to generate an updated version of the storage.ctl file before running any of the database configuration utilities that require this file as input, for example, Synchronize Schema Definition (ddlsync) and Migrate Repository (dev2prod) database configuration utilities.

An updated version of the storage.ctl file can be generated by selecting Run Database Utilities > Configure Database > Extract Storage Control File option in the Siebel Software Configuration Utility - DB Server Configuration wizard (dbsrvr_conf1 g. ksh in UNIX).

- **Method 2. Modify Generated DDL Using DB2 Tools.** Perform a standard installation (see [“Performing a Standard Installation” on page 84](#)), and choose the installation mechanism Generate DDL into Files. After the Siebel Upgrade Wizard generates the DDL (to a file named schema.sql), your DBA can directly edit the output DDL using native DB2 tools.

Some DBAs prefer using Method 2 because it only requires use of customary DB2 tools. This method does not require manipulation of the storage control file.

The storage control file should then be extracted to keep this in line with what is on the mainframe. This will save you time as there will be less manual work needed when running database utilities such as dev2prod.

You can use either of the following approaches to adjust the storage control file:

- Open the storage control file with a text editor program and edit the storage control file directly to adjust object definitions or options. This approach is useful for making minor adjustments to the database layout.

CAUTION: Do not use a text editor to edit the storage control file unless you have a thorough understanding of the storage control file structure.

- Use a Siebel-provided tool called the Siebel Database Storage Configurator (dbconf.xls) to view and edit storage control files and templates. This approach provides flexibility and ease of customization to DBAs who are customizing and managing Siebel DB2 object definitions. For more information about this tool, see ["About the Siebel Database Storage Configurator."](#)

NOTE: If your installation is in UNIX, you must transfer dbconf.xls and the storage control file that you want to configure to your Microsoft Windows environment. Use BINARY FTP to transfer the dbconf.xls file.

About the Siebel Database Storage Configurator

The Siebel Database Storage Configurator is a Microsoft Excel macro tool that allows you to import and edit attributes from one of the Siebel-provided storage control file templates. After you customize the template and save the resulting storage control file with a new name, the storage control file contains the actual names for the DB2 objects.

The file associated with the Siebel Database Storage Configurator is called dbconf.xls. This file is installed in the db2390 subdirectory of your *DBSRVR_ROOT* directory. In order to open dbconf.xls, you must have Microsoft Excel installed on your workstation.

TIP: Append the name of your new storage control file with a suffix; for example, if you used storage_p.ctl for your input, name the resulting file storage_p_suffix.ctl, where you replace *suffix* with an identifiable character string.

6

Understanding Siebel Partitioning

This section offers guidelines for partitioning large tables in Siebel eBusiness Applications. Partitioning a DB2 table is not a complicated procedure, but in order to optimally partition Siebel tables, it is critical that you understand how partitioning works in DB2. You should understand the contents of columns used to partition tables, and you should understand how ROW_IDs are used in partitioning.

This chapter consists of the following topics:

- [“About Siebel Partitioning”](#)
- [“Partitioning Strategy for Storage Control Files” on page 56](#)
- [“About Configuring Siebel Templates for Partitioning” on page 57](#)
- [“About Partitioning Tables” on page 58](#)
- [“About Methods Used for Partitioning” on page 58](#)
- [“Strategies for Special Types of Tables” on page 63](#)
- [“About Prepartitioned Siebel Tables” on page 65](#)

About Siebel Partitioning

Partitioning table spaces on DB2 allows tables to be spread across multiple physical partitions based on a partitioning key, a partitioning index, and a set of key value ranges for each partition. Using partitioned table spaces increases the maximum size of a table and improves the manageability of large tables.

Partitioning EIM table spaces by batch number improves EIM performance for batching and parallel processing, and allows distribution of key ranges across multiple data sets.

Any table, including any of the high-volume tables listed in [Table 8 on page 41](#), can be partitioned during the installation or upgrade process. You can define partitioned table spaces and key ranges for Siebel tables during or after installation, based on your business requirements. For a complete list of prepartitioned Siebel tables, see [“About Prepartitioned Siebel Tables” on page 65](#).

You can choose to partition tables yourself by following Siebel guidelines, or you can take advantage of the default partitioning scheme that Siebel Systems developed, based on Siebel customer experience using the Siebel data model with DB2 UDB for z/OS and OS/390. If you use the Siebel default partitions, you can either accept them as-is, or you can reconfigure them to suit your requirements.

When a table is created on a partitioned table space, the table is assigned a partitioning key that is composed of one or more columns. Value ranges are assigned to each partition based on value ranges within the partitioning key. The value ranges determine which partition a particular row is assigned to.

The partitioning index specifies the key values that determine how data is partitioned. You can divide a large table such as S_ORG_EXT (which holds a list of new accounts) to store records of names beginning with letters A-L in one partition and records of names beginning with letters M-Z in another partition, as shown in Figure 2. The partitioning index defines the key ranges for the partitioning column. A key range must be specified for each partition.

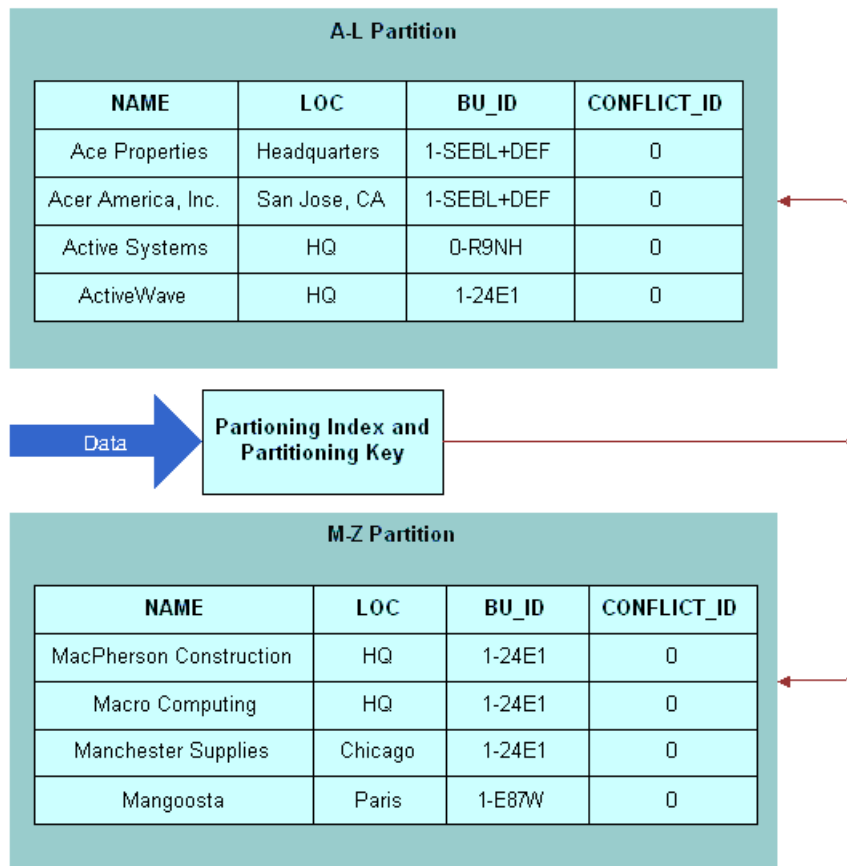


Figure 2. Partitioning S_ORG_EXT Table on DB2 UDB for z/OS and OS/390

NOTE: For more information about partitioning table spaces on DB2, consult your IBM documentation.

Partitioning Strategy for Storage Control Files

Partitioning is defined in Siebel storage control files. The following sample storage control files contain definitions for partitioning of Siebel tables:

- **storage_p.ctl.** This template is the database storage layout for Siebel Schema with partitioning for ASCII encoding. Every nonpartitioned table resides in its own segmented table space. Each table space resides in its own database.
- **storage_p_e.ctl.** This template is the database storage layout for Siebel Schema with partitioning for EBCDIC encoding. Every nonpartitioned table resides in its own segmented table space. Each table space resides in its own database.

The default partitioning scheme is optimized for online activities from the Siebel Client. Storage control files are described in detail in [Chapter 5, “Configuring the Database Layout.”](#)

Requirements for which Siebel tables to partition and how to partition them are determined by the use and the size of tables in your deployment. Careful planning, requirements analysis, and monitoring are necessary to achieve optimal partitioning. You also must monitor access path and data distribution, because changes might occur that require you to repartition table spaces.

There are no restrictions on when you can define partitioning. You can define partitioning during installation, or you can wait until you plan to use partitioning.

About Choosing a Partitioning Key for a Siebel Table

Siebel Systems supports the use of any column as the partitioning key. Based on data access and data distribution, you can choose which column is best for partitioning in your implementation. Likewise, you should determine the number of partitions required in your implementation.

NOTE: When choosing a partitioning key for a Siebel table, follow all rules, restrictions, and concerns listed in IBM DB2 documentation. These considerations include data distribution and potential updates of partitioning keys.

About Using a Partitioning Index

The partitioning index is by definition the clustering index in DB2. Therefore, by optimizing your clustering order, you optimize performance. Make sure your partitioning strategy supports more than day-to-day online access to the Siebel tables. Consider processes such as EIM, Siebel Remote, Assignment Manager, and Workflow. See [“Strategies for Special Types of Tables” on page 63.](#)

About Configuring Siebel Templates for Partitioning

Two of the storage control templates shipped with your Siebel eBusiness Applications are used for partitioning tables. The templates provide a set of 17 partitioned base tables for the Horizontal product line and a set of 24 partitioned base tables for Siebel Industry Applications.

You can identify the templates for partitioning by the characters `_p` in the name of the `.ctl` file name. (If the characters `_e` appear at the end of the file name, then the template is for the EBCDIC code page encoding scheme.) Both partitioning templates contain the same Siebel-recommended partitioning schema.

The Siebel partitioning scheme uses two approaches. The first approach is to partition a table based on existing columns in the Siebel schema; for example, S_OPTY is partitioned by columns in the U1 index.

The second approach partitions a table using an additional partitioning column (PARTITION_COLUMN). This column is populated with data using a BEFORE INSERT trigger option to generate the partitioning value for each row based on data from another column.

About Partitioning Tables

You can partition any table according to your business requirements; you can select all the tables that Siebel Systems provides for partitioning, or you can select a subset of them.

Siebel eBusiness Applications partitioning templates provide 10 partitions for each partitioned table. You select the number of partitions for your implementation based on your business needs, and define partitioning keys based on the number of partitions.

For more information about configuring storage control files, see [“Preparing the Database Layout” on page 101](#). For information on how to size your database for small, medium, large, and extra-large installations, refer to the appropriate IBM DB2 documentation.

About Methods Used for Partitioning

Siebel Systems supports two methods for partitioning:

- Partitioning based on business data
- Partitioning using a column designed for even data distribution

Partitioning Based on Business Data

This partitioning method is based on the contents of a given column. In this case, no special action is needed except to define the key ranges and number of partitions.

Partitioning for Even Data Distribution

This partitioning method involves adding a column designed specifically for Siebel partitioning with even data distribution. Siebel eBusiness Applications recognize this column by the name PARTITION_COLUMN. This column is populated by a DB2 before insert trigger.

Most Siebel tables are tied together by the ROW_ID column from a parent table. Columns with an _ID suffix are used to define a parent-child relationship, for example, OU_ID. These columns might seem to be good partitioning candidates because they support the DB2 access path; but in fact, these columns are poor candidates for partitioning because ROW_IDs are generated in sequential order.

To resolve the limitation caused by sequential order of ROW_ID, Siebel Systems provides a column, named PARTITION_COLUMN, that is based on the ROW_ID. This column is used only for partitioning purposes. It is a physical column defined in a storage control file, but it is not a part of the Siebel repository.

It is important that you continue to use the name `PARTITION_COLUMN` in order for the Siebel application to recognize this column. Also, you must define this column as `NOT NULL WITH DEFAULT`.

A DB2 before insert trigger for populating `PARTITION_COLUMN` values is defined in the Siebel storage control file. The function of this trigger is to extract the last two bytes of a source `_ID` column and use it to populate `PARTITION_COLUMN`. By using a trigger to populate the columns, you can partition tables that do not have good candidate columns for a partitioning key due to their data content. Using a trigger, you can still generate values that distribute the data well.

Scenario for Partitioning

If you were using table `S_ADDR_ORG`, the column `OU_ID` might appear to be a good candidate to define a clustering or partitioning index. However, `OU_ID` contains data in the Siebel row ID format. Row IDs are generated in ascending order, but a trigger extracts the last two bytes of the `OU_ID` column and stores this value in a new physical `PARTITION_COLUMN`.

In a storage control file, the partitioned table space is defined with two types of storage control file objects: Tablespace and Tspart (table space partitions). The storage control file objects related to partitioning are:

- Table
- Tablespace
- Tspart (table space partitions)
- IndexBase (base definition of a partitioning index)
- IndexPart (index partitions)

A table definition always includes the same options, whether the corresponding table space is partitioned or not. However, the table space and index include different options if partitioned or nonpartitioned.

The following examples reflect a partitioning scenario for the `S_ADDR_ORG` table that resides in `H0401000` partitioned table space. The Siebel partitioning template used for this scenario is `storage_p.ctl`.

NOTE: The object numbers that identify the following examples might be different for your implementation.

The example for object 6074 illustrates the Table object definition for `S_ADDR_ORG`:

```
[Object 6074]
Type          = Table
Name          = S_ADDR_ORG
Database      = SI DB0401
Tablespace    = H0401000
Cl obs       = No
```

The definition for Tablespace object `H0401000` is provided in the example for object 3197:

```
[Object 3197]
Type          = Tablespace
Name          = H0401000
```

```
Database    = SI DB0401
Define      = No
Partitions  = 10
Bufferpool  = BP1
LockSize    = Page
Lockpart     = Yes
```

You can easily identify a partitioned table space by reviewing the Partitions option in the Tablespace object of the storage control file. If the Partitions option is greater than zero, then the table space is partitioned and requires additional objects, such as Tableparts (Tspart) and partitioning index. The number of Tspart objects is equal to the number of partitions. In the example for object 3197, illustrated above, Partitions=10; therefore, ten Tspart objects must be defined (as shown in the example for objects 3198 through 3208).

```
[Object 3198]
Type        = Tspart
Name        = H0401000
PartNum     = 1
...
[Object 3208]
Type        = Tspart
Name        = H0401000
PartNum     = 10
```

The storage control file definitions for the partitioned table space in the previous example result in the following output DDL statements:

```
CREATE, TABLESPACE H0401000 IN SI DB0401 USING STOGROUP SYSDEFLT
PRIQTY 48 SECQTY 1440 FREEPAGE 4 PCTFREE 30 LOCKPART YES DEFINE NO
NUMPARTS 10
(PART 1 USING STOGROUP SYSDEFLT PRIQTY 48 SECQTY 1440 FREEPAGE 4 PCTFREE 30 COMPRESS NO ,
PART 2 USING STOGROUP SYSDEFLT PRIQTY 48 SECQTY 1440 FREEPAGE 4 PCTFREE 30 COMPRESS NO ,
PART 3 USING STOGROUP SYSDEFLT PRIQTY 48 SECQTY 1440 FREEPAGE 4 PCTFREE 30 COMPRESS NO ,
PART 4 USING STOGROUP SYSDEFLT PRIQTY 48 SECQTY 1440 FREEPAGE 4 PCTFREE 30 COMPRESS NO ,
PART 5 USING STOGROUP SYSDEFLT PRIQTY 48 SECQTY 1440 FREEPAGE 4 PCTFREE 30 COMPRESS NO ,
PART 6 USING STOGROUP SYSDEFLT PRIQTY 48 SECQTY 1440 FREEPAGE 4 PCTFREE 30 COMPRESS NO ,
PART 7 USING STOGROUP SYSDEFLT PRIQTY 48 SECQTY 1440 FREEPAGE 4 PCTFREE 30 COMPRESS NO ,
PART 8 USING STOGROUP SYSDEFLT PRIQTY 48 SECQTY 1440 FREEPAGE 4 PCTFREE 30 COMPRESS NO ,
PART 9 USING STOGROUP SYSDEFLT PRIQTY 48 SECQTY 1440 FREEPAGE 4 PCTFREE 30 COMPRESS NO ,
PART 10 USING STOGROUP SYSDEFLT PRIQTY 48 SECQTY 1440 FREEPAGE 4 PCTFREE 30 COMPRESS NO
) BUFFERPOOL BP1 LOCKSIZE PAGE LOCKMAX 0
COMPRESS NO
/
```

A partitioning index definition in the storage control file requires two types of objects: IndexBase and IndexPart (index partitions). The new partitioning index S_ADDR_ORG_P99, containing PARTITION_COLUMN, is defined in the storage control file for S_ADDR_ORG.

```
[Object 8386]
Type        = IndexBase
Name        = S_ADDR_ORG_P99
Table       = S_ADDR_ORG
Partitions  = 10
```

```

Special Col = PARTITION_COLUMN WCHAR(2) NOTNULL DEFAULT ' '
Function    = WHEN (N.OU_ID IS NOT NULL) BEGIN ATOMIC SET N.PARTITION_COLUMN =
RIGHT(N.OU_ID, 2); END
Column 1    = PARTITION_COLUMN ASC
Column 2    = OU_ID ASC

```

The example for object 8386 contains a physical partitioning column and a corresponding DB2 before insert trigger. These are always defined by the SpecialCol keyword in an IndexBase section. The syntax for the trigger implements the column PARTITION_COLUMN to S_ADDR_ORG as CHAR(2) NOT NULL with default of a space. It also implements a partitioning index and a trigger to populate the partitioning column.

In the example for object 8386, the IndexBase section defines ten (10) partitions. Therefore, it requires ten IndexPart objects. These are illustrated in the example for objects 8387 through 8396:

```

[Object 8387]
Type      = IndexPart
Name      = S_ADDR_ORG_P99
Table     = S_ADDR_ORG
PartNum   = 1
LimitKey  = '3'
...
[Object 8396]
Type      = IndexPart
Name      = S_ADDR_ORG_P99
Table     = S_ADDR_ORG
PartNum   = 10
LimitKey  = X'FF'

```

The storage control file definitions in the previous example result in the following output DDL statements:

```

CREATE INDEX SIEBTO.S_ADDR_ORG_P99 ON SIEBTO.S_ADDR_ORG
(PARTITION_COLUMN, OU_ID)
USING STOGROUP SYSDEFLT PRIQTY 48 SECQTY 1440 PCTFREE 30 DEFINE NO
CLUSTER (
PART 1 VALUES ('3') USING STOGROUP SYSDEFLT PRIQTY 48 SECQTY 1440
PCTFREE 30 FREEPAGE 4 ,
PART 2 VALUES ('7') USING STOGROUP SYSDEFLT PRIQTY 48 SECQTY 1440
PCTFREE 30 FREEPAGE 4 ,
PART 3 VALUES ('B') USING STOGROUP SYSDEFLT PRIQTY 48 SECQTY 1440
PCTFREE 30 FREEPAGE 4 ,
PART 4 VALUES ('F') USING STOGROUP SYSDEFLT PRIQTY 48 SECQTY 1440
PCTFREE 30 FREEPAGE 4 ,
PART 5 VALUES ('I') USING STOGROUP SYSDEFLT PRIQTY 48 SECQTY 1440
PCTFREE 30 FREEPAGE 4 ,
PART 6 VALUES ('N') USING STOGROUP SYSDEFLT PRIQTY 48 SECQTY 1440
PCTFREE 30 FREEPAGE 4 ,
PART 7 VALUES ('R') USING STOGROUP SYSDEFLT PRIQTY 48 SECQTY 1440
PCTFREE 30 FREEPAGE 4 ,
PART 8 VALUES ('V') USING STOGROUP SYSDEFLT PRIQTY 48 SECQTY 1440
PCTFREE 30 FREEPAGE 4 ,
PART 9 VALUES ('Z') USING STOGROUP SYSDEFLT PRIQTY 48 SECQTY 1440

```

```

PCTFREE 30 FREEPAGE 4 ,
PART 10 VALUES (X'FF') USING STOGROUP SYSDEFLT PRIQTY 48 SECQTY 1440
PCTFREE 30 FREEPAGE 4
) BUFFERPOOL BP2
/
CREATE TRIGGER SIEBTO.PTH0401 NO CASCADE BEFORE INSERT ON
SIEBTO.S_ADDR_ORG REFERENCING NEW AS N FOR EACH ROW MODE DB2SQL
WHEN (N.OU_ID IS NOT NULL) BEGIN ATOMIC SET N.PARTITION_COLUMN =
RIGHT(N.OU_ID, 2); END
/

```

The trigger name, PTH0401, is based on a number assigned to the S_ADDR_ORG table within the Siebel repository. The S_ADDR_ORG table is defined in table space H0401000. The trigger name and the table space name are both based on the number assigned to the S_ADDR_ORG table and stored in the table Group Code in the Siebel repository.

Notice that the source column OU_ID is added as the second column of the index. This makes sure that related records (records with same OU_ID) are inserted and stored in physical clustered order. If the original clustering index contains more than one column, then these too could be added to the partitioning index to sustain clustering order. In order to sustain clustering order, it is recommended that you schedule routine Reorgs of the table space and index space. Regularly scheduled Reorgs accommodate insert activity and reclaim PCTFREE and FREEPAGE definitions.

CAUTION: For most processes, the Siebel Row ID is generated as a BASE 36 value that contains the alphabetical characters A through Z and the numeric characters zero through nine (0 - 9). The EIM process uses a suffix that contains numeric characters zero through nine, so consider this when you define the key ranges.

When you are defining key ranges, remember that EBCDIC and ASCII sorting orders are different. Numeric characters precede alphabetical characters in ASCII, but it is the opposite in EBCDIC.

About the Performance Overhead of Using a Trigger

The performance overhead of using a trigger is measured as a percentage of the inserts. Performance overhead depends on several factors, including row length and the number of indexes in the table.

Triggers that use the S_PARTY partitioned table space, 10 partitions, 14 columns, and 443 bytes with 8 indexes, have demonstrated a performance overhead of less than one percent per insert.

About Changing Partitioning Keys

Siebel eBusiness Applications provide generic values for the partitioning keys. You must review these LimitKey values and change them to suit the special requirements of your implementation.

The index partition is represented in the storage control file by the IndexPart object, for example:

```

[Object 8389]
Type      = IndexPart
Name      = S_ADDR_ORG_P99
Table     = S_ADDR_ORG
PartNum   = 3
LimitKey  = 'B'

```

The number of such IndexPart objects is the same as the number of partitions shown in the IndexBase object of the same Name.

CAUTION: The sorting order for EBCDIC values is different from ASCII.

Strategies for Special Types of Tables

This section explains how to partition tables to support Siebel business processes, such as Siebel Remote, Siebel Assignment Manager, Siebel Workflow, and Siebel Enterprise Integration Manager (EIM).

About the Siebel Remote Transaction Logging Table

Siebel Remote uses the S_DOCK_TXN_LOG table for transaction logging. All user changes on the Siebel Server are logged in this table. Changes are then routed to mobile users according to their responsibilities and user privileges.

TIP: If you do not have mobile users or if you intend to use your mobile users for extract only, it is recommended that you turn off transaction logging to improve performance.

To turn off transaction logging using Siebel Tools

- 1 Choose Screens > System Administration > System Preferences.
- 2 In the System Preference column, select Docking:Transaction Logging.
- 3 Change the value in the System Preference Value column to FALSE.

Performance Considerations for Data Sharing Environment

To reduce the performance impact on the coupling facility in a data sharing environment, partition table S_DOCK_TXN_LOG by MEMBER_NAME.

To avoid giving multiple members ReadWrite access to the same partition, create a column called MEMBER_NAME that is populated by the CURRENT MEMBER special register, thereby verifying that all write processes associated with the member affect only one partition.

Partitioning by MEMBER_NAME is not a default, because such partitioning is only required for a data sharing environment. Therefore, the names of the data sharing members are not known by the Siebel application. Create the new partitioning index using the syntax in the following example.

```
[Object 8837]
Type      = IndexBase
Name      = S_DOCK_TXN_LOG_PART
Table     = S_DOCK_TXN_LOG
Special Col = MEMBER_NAME WCHAR(8)
```

Partitioning S_ESCL_REQ and S_ESCL_LOG Tables

Tables S_ESCL_REQ and S_ESCL_LOG, which are used by Assignment Manager and Siebel Business Process Designer, tend to grow very large. S_ESCL_REQ is processed based on the GROUP_ID, whereas S_ESCL_LOG is processed by RULE_ID.

Partitioning of these tables can be done in one of the following ways:

- **Hard-coding the actual values for GROUP_ID and RULE_ID.** This approach is recommended when the number of groups is static. Additional partitions can be defined to accommodate new values that are unknown at creation time. To use this option, you must first define the groups and then extract the actual GROUP_ID values you want to use for partitioning.
- **Adding the PARTITION_COLUMN and a trigger to populate the column.** The number of partitions defined should ensure that data from multiple groups goes into multiple partitions. Start with 36 partitions, because this is the base number for the Siebel row ID.

NOTE: If you encounter resource contention due to multiple groups using the same partition when you create new groups and assign new group IDs, consider increasing the number of partitions to allow for a more granular level of GROUP_ID assignments to a partition range.

Partitioning EIM Tables for Performance

To speed up EIM load and reduce I/O (input and output) contention, make sure all partitions for EIM tables are spread across the entire I/O subsystem. You can use the Siebel Storage Configuration process to define partitions on EIM tables. For detailed information about this process, see [Chapter 5, “Configuring the Database Layout.”](#)

NOTE: The sample storage control files also contain partitioning of certain EIM tables. Before you use EIM, verify that you are using the correct key ranges, because key ranges depend on which batch numbers are used.

It is recommended that you partition EIM tables based on _U1 indexes of EIM tables, that is, IF_ROW_BATCH_NUM, ROW_ID. This allows an EIM batch input to be assigned to one partition, thereby allowing multiple EIM batches to be run in parallel.

You can compute the optimal number of partitions for an EIM table by dividing the number of rows in the EIM table by the number of parallel processes you intend to run. The result should be the approximate number of partitions for that table.

The recommended partitioning index is usually the U1 index, which is IF_ROW_BATCH_NUM and ROW_ID. In some cases, you might need to consider a different index for partitioning.

In the following example, EIM_CONTACT is partitioned based on IF_ROW_BATCH_NUM and CON_LAST_NAME. The target base table S_CONTACT is also partitioned based on LAST_NAME. This results in less random I/O by DB2 while accessing the S_CONTACT table.

```
CREATE INDEX SIEBTO.EIM_CONTACT_P99
ON SIEBTO.EIM_CONTACT
( IF_ROW_BATCH_NUM      ASC ,
  CON_LAST_NAME         ASC )
```


This special method applies only to target base tables that are partitioned based on the natural key—not by `PARTITION_COLUMN`, which is derived from `ROW_ID`. The following examples illustrate partitioning EIM tables for `S_ORG_EXT` and `S_OPTY`.

```
CREATE INDEX SIEBTO.EIM_ACCOUNT_P99
ON SIEBTO.EIM_ACCOUNT
( IF_ROW_BATCH_NUM ASC ,
  NAME ASC ,
  ACCNT_BI ASC ,
  LOC ASC )
```

```
CREATE INDEX SIEBTO.EIM_OPTY_P99
ON SIEBTO.EIM_OPTY
( IF_ROW_BATCH_NUM ASC ,
  OPTY_NAME ASC )
```

About EIM Considerations for Table Space Partitioning

The mechanism by which EIM generates the `ROW_ID` can result in an uneven distribution of data if you use EIM with one of the default partitioning schemes to import data into a base table.

When EIM imports data into a base table, EIM automatically generates the base table `ROW_ID` by concatenating the prefix with a unique number; for example, 1-SB3-123. EIM relies on database functions to make sure that each number is unique under the same prefix.

Uneven data distribution occurs because one of the default partitioning approaches uses the last two characters of `ROW_ID` as the partition key for partitioning of base tables; these characters can be letters, numbers, or a combination of letters and numbers, but the last two characters in a `ROW_ID` generated by EIM are *numbers*.

In order for your storage control file to use partitioning keys that reflect the nature of your data, consider your data distribution with regard to your EIM process before you begin your database installation. Then modify your partitioning keys accordingly. Alternatively, you can repartition the table after an uneven distribution has occurred.

About Prepartitioned Siebel Tables

Siebel eBusiness Applications Release 7.7 provides the following partitioned tables in storage control file templates.

The tables identified in [Table 12](#) represent partitioned Core product tables; seven exceptions are identified as Siebel Industry Applications (SIA) product tables.

Table 12. Prepartitioned Siebel Tables with Physical Columns

Table	Index	Indexed Columns	Trigger Logic
S_ACT_EMP	S_ACT_EMP_P99	PARTITION_COLUMN, EMP_ID	" WHEN (N.IMP_ID IS NOT NULL) BEGIN ATOMIC SET N. PARTITION_COLUMN = RIGHT(N. EMP_ID, 2); END →"
S_ADDR_ORG	S_ADDR_ORG_P99	PARTITION_COLUMN, OU_ID	" WHEN (N.OU_ID IS NOT NULL) BEGIN ATOMIC SET N. PARTITION_COLUMN = RIGHT(N. OU_ID, 2); END →"
S_ADDR_PER	S_ADDR_PER_P99	PARTITION_COLUMN, PER_ID	" WHEN (N.PER_ID IS NOT NULL) BEGIN ATOMIC SET N. PARTITION_COLUMN = RIGHT(N. PER_ID, 2); END →"
S_APPLD_CVRG Siebel Industry Applications (SIA) table	S_APPLD_CVRG_P99	PARTITION_COLUMN, ASSET_ID	WHEN (N. ASSET_ID IS NOT NULL) BEGIN ATOMIC SET N. PARTITION_COLUMN = RIGHT(N. ASSET_ID, 2); END →"

Table 12. Prepartitioned Siebel Tables with Physical Columns

Table	Index	Indexed Columns	Trigger Logic
S_ASSET	S_ASSET_P99	PARTITION_COLUMN, ROW_ID	"WHEN (N. ROW_ID IS NOT NULL) BEGIN ATOMIC SET N. PARTITION_COLUMN = RIGHT(N. ROW_ID, 2); END →" S_ASSET
S_ASSET_CON	S_ASSET_CON_P99	PARTITION_COLUMN,ASSET_ID	"WHEN (N. ASSET_ID IS NOT NULL) BEGIN ATOMIC SET N. PARTITION_COLUMN = RIGHT(N. ASSET_ID, 2); END →"
S_ASSET_POSTN Siebel Industry Applications (SIA) table	S_ASSET_POSTN_P99	PARTITION_COLUMN,ASSET_ID	"WHEN (N. ASSET_ID IS NOT NULL) BEGIN ATOMIC SET N. PARTITION_COLUMN = RIGHT(N. ASSET_ID, 2); END →"
S_COMMUNICATION	S_COMUNICATION_P99	PARTITION_COLUMN,ROW_ID	"WHEN (N. ROW_ID IS NOT NULL) BEGIN ATOMIC SET N. PARTITION_COLUMN = RIGHT(N. ROW_ID, 2); END →"
S_CONTACT	S_CONTACT_M12	LAST_NAME, FST_NAME, PRIV_FLG, ALIAS_NAME	N/A

Table 12. Prepartitioned Siebel Tables with Physical Columns

Table	Index	Indexed Columns	Trigger Logic
S_EVT_ACT	S_EVT_ACT_P99	PARTITION_COLUMN,OPTY_ID	"WHEN (N. ROW_ID IS NOT NULL) BEGIN ATOMIC SET N. PARTITION_COLUMN = RIGHT(N. OPTY_ID, 2); END →"
S_EXP_ITEM	S_EXP_ITEM_P99	PARTITION_COLUMN, OWNER_EMP_ID	"WHEN (N. OWNER_EMP_ID IS NOT NULL) BEGIN ATOMIC SET N. PARTITION_COLUMN = RIGHT(N. OWNER_EMP_ID, 2) ; END →"
S_FN_ACCNT_TXN Siebel Industry Applications (SIA) table	S_FN_ACCNT_TXN_P99	PARTITION_COLUMN, ACCNT_ID	"WHEN (N. ACCNT_ID IS NOT NULL) BEGIN ATOMIC SET N. PARTITION_COLUMN = RIGHT(N. ACCNT_ID, 2); END →"
S_FNCVRG_ELMNTS Siebel Industry Applications (SIA) table	S_FNCVRG_ELMNTSP99	PARTITION_COLUMN,APPLD_CVR G_ID	"WHEN (N. APPLD_CVRG_ID IS NOT NULL) BEGIN ATOMIC SET N. PARTITION_COLUMN = RIGHT(N. APPLD_CVRG_ID, 2) ; END →"

Table 12. Prepartitioned Siebel Tables with Physical Columns

Table	Index	Indexed Columns	Trigger Logic
S_INS_CLAIM Siebel Industry Applications (SIA) table	S_INS_CLAIM_P99	PARTITION_COLUMN, INSCCLAIM_NUM	"WHEN (N. INSCCLAIM_NUM IS NOT NULL) BEGIN ATOMIC SET N. PARTITION_COLUMN = RIGHT(N. INSCCLAIM_NUM, 2) ; END →"
S_INS_ITEM Siebel Industry Applications (SIA) table	S_INS_ITEM_P99	PARTITION_COLUMN, ASSET_ID	"WHEN (N. ASSET_ID IS NOT NULL) BEGIN ATOMIC SET N. PARTITION_COLUMN = RIGHT(N. ASSET_ID, 2); END →"
S_INSCLM_ELMNT Siebel Industry Applications (SIA) table	S_INSCLM_ELMNT_P99	PARTITION_COLUMN, INSCLM_ID	"WHEN (N. INSCLM_ID IS NOT NULL) BEGIN ATOMIC SET N. PARTITION_COLUMN = RIGHT(N. INSCLM_ID, 2); END →"
S_OPTY	S_OPTY_U1	NAME, PR_DEPT_OU_ID, BU_ID, CONFLICT_ID	N/A
S_OPTY_POSTN	S_OPTY_POSTN_P99	PARTITION_COLUMN, OPTY_ID	"WHEN (N. OPTY_ID IS NOT NULL) BEGIN ATOMIC SET N. PARTITION_COLUMN = RIGHT(N. OPTY_ID, 2); END →"
S_ORG_EXT	S_ORG_EXT_U1	NAME, LOC, BU_ID, CONFLICT_ID	N/A

Table 12. Prepartitioned Siebel Tables with Physical Columns

Table	Index	Indexed Columns	Trigger Logic
S_PARTY	S_PARTY_P99	PARTITION_COLUMN,ROW_ID	<pre> "WHEN (N. ROW_ID IS NOT NULL) BEGIN ATOMIC SET N. PARTITION_COLUMN = RIGHT(N. ROW_ID, 2); END →" </pre>
S_PARTY_REL	S_PARTY_REL_P99	PARTITION_COLUMN,PARTY_ID	<pre> "WHEN (N. PARTY_ID IS NOT NULL) BEGIN ATOMIC SET N. PARTITION_COLUMN = RIGHT(N. PARTY_ID, 2); END →" </pre>
S_POSTN_CON	S_POSTN_CON_P99	PARTITION_COLUMN,POSTN_ID	<pre> "WHEN (N. POSTN_ID IS NOT NULL) BEGIN ATOMIC SET N. PARTITION_COLUMN = RIGHT(N. POSTN_ID, 2); END →" </pre>
S_REVN	S_REVN_P99	PARTITION_COLUMN,ROW_ID	<pre> "WHEN (N. ROW_ID IS NOT NULL) BEGIN ATOMIC SET N. PARTITION_COLUMN = RIGHT(N. ROW_ID, 2); END →" </pre>
S_SRV_REQ	S_SRV_REQ_P99	PARTITION_COLUMN,SR_NUM	<pre> "WHEN (N. SR_NUM IS NOT NULL) BEGIN ATOMIC SET N. PARTITION_COLUMN = RIGHT(N. SR_NUM, 2); END →" </pre>

7

Installing the Siebel Database Server Files at Midtier

This chapter describes file installation at midtier on both the Windows and UNIX platforms and consists of the following topics:

- [“About File Installation at Midtier”](#)
- [“About the Siebel Database Server” on page 72](#)
- [“Preinstallation Tasks for the Siebel Database Server” on page 72](#)
- [“Siebel Database Server Software Installation” on page 73](#)
- [“Reviewing the Software Installation for DB2 UDB for z/OS and OS/390” on page 80](#)

About File Installation at Midtier

For an overview of the tasks involved in file installation at midtier, see [Table 13](#).

Table 13. Database Server Installation and Configuration Tasks

Who Performs It?	Task
Siebel administrator	Fill out your copy of Appendix A, “Deployment Planning Worksheet” with the parameter values that you need to perform the installation.
	Install DB2 Connect. See Chapter 4, “Preparing for Implementation on the DB2 Host.”
	Install any required Fix Packs. See <i>System Requirements and Supported Platforms</i> on Siebel SupportWeb.
Database administrator (DBA)	Create the DB2 subsystem for Siebel eBusiness Applications. See Chapter 4, “Preparing for Implementation on the DB2 Host.”
	Allocate and configure disk space appropriate to your installation requirements. See Chapter 4, “Preparing for Implementation on the DB2 Host.”
Siebel administrator and DBA	Install the Siebel Database Server software. See “Siebel Database Server Software Installation” on page 73 .
	Review the software installation. See “Reviewing the Software Installation for DB2 UDB for z/OS and OS/390” on page 80 .

About the Siebel Database Server

The Siebel Database Server stores the data used by Siebel eBusiness Applications. Siebel Dedicated Web Clients, Siebel Tools Clients, and Siebel Server components connect directly to the Siebel Database Server and make changes in real time. Dedicated Web Clients download a subset of the data to use locally, periodically synchronizing with the Database Server through the Siebel Server to update both.

Installation of the Siebel Database Server software provides files with which to install Siebel Database and Siebel Database Server components and to import the Siebel Repository.

The Siebel Web Client connects to the Siebel Server, which accesses the Siebel Database.

Preinstallation Tasks for the Siebel Database Server

Before installing the Siebel Database Server, make sure that you have the following resources available to you for installation:

- Your system administrator (SYSADM) and database administrator (DBA) have already set up your DB2 subsystem as a precondition for installing your database server.
- The SYSADM has already created storage groups and activated and granted use of buffer pools in preparation for installation.
- A security administrator (or someone with a logon that can create user IDs and security groups) is available to support your installation.
 - This administrator must create the user ID that will act as the database user (for example, SADMIN). The database user should be a member of the security group with authorization to set the current SQLID to the schema name.
 - The security administrator must also create the groups needed for general Siebel application users (for example, SSEROLE).

For more details regarding authorization requirements, see [Chapter 3, "Security Concepts for z/OS and OS/390."](#)

- A database user (for example, SADMIN) with DBA/CREATEDBA authority exists. This person must also have TSO logon capability for the manual installation process.
- The Database Owner is a valid authorization ID.
- If you have not already done so, make a copy of the Deployment Planning Worksheet, located in [Appendix A, "Deployment Planning Worksheet,"](#) fill it out, and distribute copies to the entire installation team.

To prepare for the Siebel Schema installation

- 1 Confirm that you have DB2 V7 installed and that you have run the DB2 Installation Verification Program (IVP).

- 2 Follow the applicable instructions regarding setting up your DB2 Subsystem, as described in [Chapter 4, "Preparing for Implementation on the DB2 Host."](#) This chapter includes installation requirements such as APARs, DB2 work files, disk space, buffer pools, and storage groups.
- 3 Install DB2 Connect, as described in [Chapter 4, "Preparing for Implementation on the DB2 Host,"](#) and test ODBC connectivity between the machine on which the Siebel Server will reside and DB2.
- 4 Review the *Siebel Installation Guide* for the operating system you are using. Install the Siebel Gateway Name Server, the Siebel Enterprise Server, and the Siebel Server software in a base language and any additional languages your enterprise requires.
- 5 Designate a Siebel Server to act as a client to DB2.

This server is where you will locate the Siebel Database Server directory.

NOTE: Verify that all the Siebel-supported languages you need are installed on the DB2-client Siebel Server. Otherwise, the installation program will not recognize the additional languages when you add them to the Siebel Schema.

- 6 (Optional) Install Microsoft Excel on the client machine to allow use of the Siebel Database Configurator tool.

Siebel Database Server Software Installation

Installation of Siebel Database Server software consists of the following tasks:

- ["Installing the Database Server Software Under Windows"](#)
- ["Installing the Database Server Software Under UNIX" on page 76](#)

CAUTION: To install Siebel eBusiness Applications, you must have between 50 and 100 MB of disk space on your system drive, even if you intend to install Siebel eBusiness Applications on another drive.

Installing the Database Server Software Under Windows

Complete the steps below to install the Siebel Database Server files under Windows. You must already have a Siebel Server installed on the computer.

To install the database server software under Windows

- 1 Insert the *Siebel eBusiness Applications, Base Applications for Windows* DVD into the DVD drive of the chosen server.
- 2 In Windows Explorer, double-click *seawi nuni base* as appropriate to the Siebel applications you are installing.

- 3 Navigate to Windows\Server\Siebel_Enterprise_Server and double-click setup.exe.
The Welcome to InstallShield Wizard for Siebel Enterprise Server screen appears.
- 4 Read the Welcome to the InstallShield Wizard for Siebel Enterprise Server screen and click Next to continue with the setup program.

Check to be sure there are no other programs open before continuing.

If this is the first time you are installing, on the next screen you will select the products you wish to install. Proceed to [Step 6](#).

If this is not the first time you are installing, setup will detect an existing version of this software on your system.
- 5 When an existing version of the software is detected on your system, select from one of the following two options:
 - To install this server, select None to install a new server and click Next. Proceed to [Step 6](#).
 - To install a new language pack to an existing installation instance, select the instance and click Next. Proceed to [Step 10 on page 75](#).
- 6 The installer installs the Siebel Database Server software in the default directory C:\sea7xx. If you wish to choose a different installation directory:
 - a Click Browse.
 - b Enter the name of the directory you recorded in [Appendix A, "Deployment Planning Worksheet."](#)

NOTE: This directory name must not contain spaces, apostrophes, hyphens, or other special characters. Underscores are allowed.

NOTE: You must install the Siebel Database Server in the same directory in which you installed the Siebel Server.
The installer prompts you to select the products that you want to install.
- 7 Choose the products to install from the following options:
 - Select all components for which you have a license.
 - Select just the Siebel Database Server at this time for installation and configuration.
CAUTION: If you choose to install and configure all the components at once for which your company has a license, you cannot later uninstall any of them individually. You must uninstall them all as a group at the same time.

If you install all licensed components at once, the SES Installer and the Siebel Software Configuration Utility prompt you for the installation parameters of each component individually and in the sequence required *except in the case of the database server*. You must activate the Database Server configuration utility manually to configure the software.

To continue, click Next.

The installer prompts you to select the type of installation setup you prefer.
- 8 Choose the type of Siebel Database Server installation to execute from the following options and click Next.

- **Typical.** This setup option installs all Siebel Database Server components except those displayed. Proceed to [Step 10](#).
- **Compact.** There is no compact installation option for this server.
- **Custom.** This setup option lets you customize your installation by choosing the specific components you want to install. For example, using the Custom installation option, you can install only the software for DB2 UDB for z/OS and OS/390, rather than software for all the supported databases. Proceed to [Step 9](#).

CAUTION: If you select Custom installation, make sure that you also select Sample File Attachments the first time you install the software. These attachments must be copied to the Siebel File System directory after configuration. Otherwise, the required file attachments will not be installed.

To continue, click Next.

- 9 In the Select Components screen, select just those database platform components you want to install, in this case, DB2 UDB for z/OS and OS/390 plus the Sample File Attachments, as described earlier.

To continue, click Next.

- 10 Choose the Language Pack or Packs you are installing for the Siebel Database Server and click Next.

All servers are installed, at a minimum, with the primary language in which the server will be run. This is the primary (base) language for your Enterprise. Optionally, you can install one or more additional languages.

When installing languages at a later date, you must also reinstall any patches that have been run on the directory.

NOTE: In a Unicode-enabled database environment, you can install any of the available Siebel language packs. In a non-Unicode database environment, you must consider the correlation of the language packs you want to install and the characters supported by your database code page. For example, in a Western European code page database, you can only install Western European language packs such as English, French, Spanish, or German. In a Japanese code page database, you can only install Japanese or English language packs.

For a list of supported code pages and encoding strategies, see *System Requirements and Supported Platforms* on Siebel SupportWeb.

To continue, click Next.

- 11 When the screen appears, the installer automatically adds program icons to a folder in the Windows Program file called Siebel Enterprise Server Configuration 7.x.x after you click Next.
 - To override the default action and install the program icons to a different Program folder in the list under Existing Folders, select that folder and click Next.
 - To accept the default action, click Next.

The next screen offers you the opportunity to review the settings you have selected.

- 12 During the Siebel Database Server installation, an information screen appears stating:

Setup did not find the "Language" language pack on the current media. Please insert the DVD containing the "Language" language pack and select "setup.exe" file from the "Language" folder.

where:

Language = the Siebel code for the Language Pack you installed; for example, enu for U.S. English.

Click OK.

- 13** Remove the current DVD from the drive and insert the Siebel *Language* Language Kit DVD-ROM (where *Language* stands for the Language Pack you want to deploy, such as English for U.S. English) into the DVD-ROM drive of the same server.

- 14** In Windows Explorer, navigate to the Server\Siebel_Enterprise_Server\language directory (where *language* stands for the Language Pack you want to deploy, such as enu for U.S. English) and double-click setup.exe.

- 15** After the language pack installation is complete, an information screen appears stating:

Please re-insert the base DVD and browse to the "setup.exe" file to enable setup to continue.

Click OK.

- 16** After all Siebel Database Server files, including the language you specified, have been installed, the Event Log appears, displaying the status of your installation.

This log can also be accessed under sea7xx\SESsetup.log

To continue, click Next.

- 17** When the InstallShield Wizard Complete screen announces that software installation was completed successfully, click Finish.

A Windows Explorer window appears, showing the directory where the Database Server Configuration Utility icon resides alongside icons for any other servers you have already installed.

Proceed to ["Reviewing the Software Installation for DB2 UDB for z/OS and OS/390" on page 80.](#)

Installing the Database Server Software Under UNIX

Complete the steps below to install the Siebel Database Server files under UNIX. You must already have a Siebel Server installed on the computer.

To install the database server software under UNIX

- 1 Install the *Siebel eBusiness Applications, Base Applications for UNIX_OS* DVD onto the network.
where:

UNIX_OS is your UNIX operating system, such as Solaris, HP-UX, or AIX.

NOTE: The volume label for the DVD is *seaUNIX_OSuni base*, where *UNIX_OS* is your UNIX operating system. The volume label may not be required, depending on how you access the DVD.

As a convenience, you may also want to install the *seaall / languagelep* DVD, where *language* is the language code for the language you are planning to install, into the drive of the machine on which you want to install the Siebel Database Server.

- 2 Verify the directories and permissions to those directories into which you will install the product. You must have write and execute permission.
- 3 Navigate to the */UNIX_OS/Server/Siebel_Enterprise_Server* directory on the DVD and open a new shell.
- 4 Unset any Siebel-specific environment variables. To view current environment variable settings, enter *env* in the shell window.
- 5 To start the Siebel Database Server file installation, enter the following command:

```
./setupUNIX_OS
```

where:

UNIX_OS = the UNIX operating system, such as Solaris, HP-UX, or AIX.

The Welcome to InstallShield Wizard for Siebel Enterprise Server screen appears.

- 6 Click Next.

If you have already installed other Siebel components on the same machine, the installer displays the message that an existing installation has been found.

- 7 Depending on whether you are installing your Siebel Database Server files for the first time or adding a new language to an existing instance, take the appropriate action:
 - To install the server software in a new instance, select None as the default and click Next. Proceed to [Step 8](#).
 - To install a new language in an existing instance, select the displayed instance and click Next. Proceed to [Step 12 on page 79](#).

NOTE: You must install the Siebel Database Server in the same directory in which you installed the Siebel Server.

The Installer Path screen appears.

- 8 Enter the fully qualified path to the installation directory, and then click Next.
The installer prompts you to select the server that you want to install.
- 9 Select from the following options and click Next:

- Install all the components at once for which your organization has a license by selecting all the check boxes.
- Select just the Siebel Database Server at this time for installation and configuration. (You will install and configure the other server components individually later.)

NOTE: If you install all licensed components at once, the SES Installer and the Siebel Software Configuration Wizard prompt you for the installation parameters of each component individually and in the sequence required.

The installer prompts you to select the type of installation setup you prefer.

- 10** Choose the type of Siebel Database Server installation to execute from the following options, then click Next to continue:

- **Typical**
- **Compact** (not supported)
- **Custom**

If you chose the Custom installation type, proceed to [Step 11](#).

If you chose the Typical installation type, proceed to [Step 12 on page 79](#).

For a list of the installable components, see the following table.

Installation Setup	Products
Typical or Custom	Select the database files you want to install from the components below: <ul style="list-style-type: none">■ Sample Database support—File attachments for Siebel seed data■ Oracle Database Enterprise Edition—Database scripts for the Siebel Database on Oracle■ IBM DB2 UDB for UNIX and Windows—Database scripts for the Siebel Database on DB2 UDB for UNIX and Windows■ IBM DB2 UDB for z/OS and OS/390—Database scripts for the Siebel Database on DB2 UDB for z/OS and OS/390

- 11** Select both IBM DB2 UDB for z/OS and OS/390 and the Siebel Sample Database.

If you are installing additional languages, the Installer Language Selection screen appears.

- 12 Choose the Language Pack or Packs you are installing for the Siebel Database Server and click Next.

All servers are installed, at a minimum, with the primary language in which the server will be run. This is the primary (base) language for your Enterprise. Optionally, you can install one or more additional languages.

When installing languages at a later date, you must also reinstall any patches that have been run on the directory.

NOTE: In a Unicode-enabled database environment, you can install any of the available Siebel language packs. In a non-Unicode database environment, you must consider the correlation of the language packs you want to install and the characters supported by your database code page. For example, in a Western European code page database, you can only install Western European language packs such as English, French, Spanish, or German. In a Japanese code page database, you can only install Japanese or English language packs.

For a list of supported code pages and encoding strategies, see *System Requirements and Supported Platforms* on Siebel SupportWeb.

The installer prompts you that the setup program will install program shortcuts to your Program Folder with the name Siebel Enterprise Server 7.x.x.

- 13 Click Next and the installer automatically adds program icons to a folder in the Windows Program file called Siebel Enterprise Server 7.x.x.

NOTE: To override the default action and install the program icons to a different Program folder in the list under Existing Folders, enter the name for that folder.

The installer displays the location into which it will install the Siebel Database Server and any other servers you have elected to install. It also displays the file size.

- 14 During the Siebel Database Server installation, an information screen appears stating:

Setup did not find the "Language" Language pack on the current media. Please insert the DVD containing the Siebel *Language* and select setup*UNIX_OS* from the *Language* folder.

where:

Language = the Siebel code for the Language Pack you installed; for example, enu for U.S. English.

UNIX_OS = the UNIX operating system, such as Solaris, HP-UX, or AIX.

- 15 Remove the current DVD from the drive and insert the Siebel *Language* Language Kit DVD-ROM (where *Language* stands for the Language Pack you want to deploy, such as English for U.S. English) into the DVD-ROM drive of the same server.

A dialog box titled Navigate to the Executable on the *Language* language pack DVD appears and displays a file called siebel . i n i . However, you do not need to do anything with this file.

Clear the filter dialog box, if needed, to be sure that you can see the appropriate files.

- 16** Navigate to the `UNIX_OS\Server\Siebel_Enterprise_Server\language` directory (where *language* stands for the Language Pack you want to deploy, such as `enu` for U.S. English) and double-click `setupUNIX_OS`.

A message appears stating that language files are being installed.

- 17** When installation of all the language files has been completed, the installer prompts you again for the base DVD.

Please re-insert the base DVD and browse to the `setupUNIX_OS` file to enable setup to continue.

- 18** Navigate again to the base DVD mount point on the network, locate `Siebel_Enterprise_Server/setupUNIX_OS`, and click OK.

After all server files specified have been installed, the installer displays the message:

The InstallShield Wizard has successfully installed the Siebel Enterprise Server. Click Finish to exit the Wizard.

- 19** Click Finish.

- 20** Proceed to [“Reviewing the Software Installation for DB2 UDB for z/OS and OS/390.”](#)

Reviewing the Software Installation for DB2 UDB for z/OS and OS/390

Review the directory structure created by the Siebel Database Server installation, as illustrated below. The example below results from a custom installation with the DB2390 option only.

Windows Installation Directory for DB2 UDB for z/OS and OS/390

The DB2390 subdirectory is located under the `DBSRVR_ROOT` directory within the Siebel root directory you specified during the installation, for example, `C:\sea7xx\dbsrvr\DB2390\` under Windows.

`DBSRVR_ROOT`

```

    BIN
    COMMON
    DB2UDB
    DB2390
                                DBOUTPUT\
                                    DDLSYNC
                                    DEV2PROD
                                    INSTALL
                                    INSTALLDW
                                    UPGRADE
                                STORPROC
```


UPGRADE
LANGUAGE (for example, ENU)
DB2390
FILES
LOCALE
PE60

- **BIN.** Contains files for internal use.
- **COMMON.** Contains common database platform-independent files.
- **DB2UDB.** Contains installation and upgrade files specific to DB2 UDB.
- **DB2390.** Contains installation and upgrade files specific to DB2 UDB for z/OS and OS/390.
- **DBOUTPUT.** Contains subdirectories in which DDL has been deposited by the installer for later transfer to the DB2 host as a result of choosing Generate DDL Into Files mode.

The subdirectories to DBOUTPUT contain DDL specific to the following Siebel processes:

 - **DDLSYNC.** Related to synchronization of the Siebel Schema definition.
 - **DEV2PROD.** Related to migrating development customizations to the test or production environment.
 - **INSTALL.** Contains all installation files for the option *Generate DLL into Files*, used to create the Siebel Schema directly on the mainframe.
 - **INSTALLDW.** Contains all files necessary to install the Siebel Data Warehouse.
 - **UPGRADE.** Related to version upgrade activities.
- **STORPROC.** Contains stored procedures required for upgrades.
- **UPGRADE.** Directories containing files to allow upgrade from specific versions of Siebel eBusiness Applications supported for upgrade to the current release.
- **LANGUAGE.** Contains language- and database-specific files for the ancestor repository and supporting files. For example, ENU contains U.S. English-language-specific files, and DEU contains German language-specific files. Additional subdirectories exist under this level, based on the language packs that you install.
- **FILES.** Contains sample file attachments. These should be copied to the appropriate subdirectory within the Siebel File System. See [“Populating the Siebel File System” on page 137](#).
- **LOCALE.** Contains translation files for the Upgrade Wizard. For Siebel Systems use only.
- **PE60.** Files used for uploading data from Siebel Sales Personal Edition.

UNIX Installation Directory for DB2 UDB for z/OS and OS/390

Review the directory structure created by the Database Server installation for UNIX, as illustrated below. The example below results from a Custom installation with the db2390 option only.

`$DBSRVR_ROOT`

```
bin
common
db2390

                                dboutput/
                                    ddlsync
                                    dev2prod
                                    install
                                    installdw
                                    upgrade

                                storproc
                                upgrade

                                language (for example, enu)

                                db2390

files
locale
```

- **bin.** Contains files for internal use.
- **common.** Contains common database platform-independent files.
- **db2390.** Contains installation and upgrade files specific to DB2 UDB for z/OS and OS/390.
 - **dboutput.** Contains subdirectories in which DDL has been deposited by the installer for later transfer to DB2 host as a result of choosing Generate DDL Into Files mode.

The subdirectories below contain DDL specific to the following Siebel processes:

 - ❑ **ddlsync.** Related to synchronization of the Siebel Schema definition.
 - ❑ **dev2prod.** Related to migrating development customizations to the test or production environment.
 - ❑ **install.** Contains all installation files for the option *Generate DLL into Files*, used to create the Siebel Schema directly on the mainframe.
 - ❑ **installdw.** Contains all files necessary to install the Siebel Data Warehouse.
 - ❑ **upgrade.** Related to version upgrade activities.
 - **storproc.** Contains stored procedures required for upgrades.
 - **upgrade.** Directories containing files to enable upgrade from specific versions of Siebel eBusiness Applications supported for upgrade to the current release.
- **language.** Contains language- and database-specific files for the ancestor repository and supporting files. For example, enu contains U.S. English-language-specific files, and deu contains German-language-specific files. Additional subdirectories exist under this level, based on the language packs that you install.
- **files.** Contains sample file attachments. These should be copied to the appropriate subdirectory within the Siebel File System. See [“Populating the Siebel File System” on page 137](#).
- **locale.** Contains translation files for the Upgrade Wizard. For Siebel Systems use only.

8

Siebel Schema Installation on the DB2 Host

This chapter describes the standard and custom Siebel Schema installation and provides instructions for creating a storage control file for a custom installation.

This chapter consists of the following topics:

- [“About Installing the Siebel Schema on the DB2 Host”](#)
- [“Installing the Siebel Schema” on page 84](#)
- [“Performing a Standard Installation” on page 84](#)
- [“Preparing the Database Layout” on page 101](#)
- [“Creating and Executing the DDL” on page 113](#)
- [“Performing a Custom Installation” on page 114](#)
- [“Rerunning the Installation” on page 120](#)

About Installing the Siebel Schema on the DB2 Host

For an overview of the tasks for installing the Siebel Schema on the DB2 host, see [Table 14](#).

Table 14. Database Server Installation and Configuration Tasks

Who Performs It?	Task
Siebel administrator and database administrator (DBA)	<p>Review information about installing the tables, indexes, and seed data, using either the standard or custom option. See “Installing the Siebel Schema” on page 84.</p> <p>Install the Siebel Schema:</p> <ul style="list-style-type: none">■ If installing by standard mode, see “Performing a Standard Installation” on page 84.■ If installing by custom mode, see “Preparing the Database Layout” on page 101. Perform the custom installation using the created storage control file. For information, see “Performing a Custom Installation” on page 114. <p>Review the log files for any errors. See “Reviewing the Log Files for Windows” on page 117 or “Reviewing the Log Files for UNIX” on page 119.</p>

Installing the Siebel Schema

When you choose *Install a new Siebel database* from the Options menu of the Siebel Software Configuration Utility, the program:

- Creates Siebel tables and indexes in a specified database storage layout
- Installs Siebel triggers (optional) and views
- Installs Siebel seed data specific to your database

If you are installing to multiple environments, you must repeat the installation of database server components for each environment.

Siebel Schema creation is performed using one of two modes—standard or customized. Each mode also allows you to submit DDL automatically against the DB2 host or to apply it manually.

- **Standard Install.** If you are performing a standard installation, you perform all installation tasks using the Siebel Software Configuration Utility—DB Server Configuration wizard. This will create a Siebel out-of-the-box database layout.
- **Customized Install.** If you are performing a custom installation, you will need to provide a storage control file, which was created and validated prior to performing this installation. This will allow you to create a custom database layout, best suited to your Siebel deployment.

NOTE: The Upgrade Wizard is a Siebel utility that is used for database server installation, upgrade, and database manipulations. The utility is launched by the Configure DB Server wizard. In Windows, the Siebel Upgrade Wizard is designated as `si ebupg.exe`. In UNIX, it is designated as `si ebupgwi z`.

Installing a New Database Language

Each time you install a new database language, you must reinstall the Siebel Database Server components by launching the Database Server configuration wizard and choosing Siebel Database Server Options > Install > Add a language to existing database.

CAUTION: If you add a new language *after* you have completed your initial installation with your initial set of languages, when the configuration utility launches, leave the Gateway Name Server Address and Enterprise Server Name fields blank. If you enter values in these fields, an error will occur.

Performing a Standard Installation

This procedure describes how to perform a standard installation. You can use the standard install option as your sole installation procedure, or you can perform the standard installation as a first step of a custom installation and then customize your deployment after you have completed the standard installation. To perform a custom installation, see [“Performing a Custom Installation” on page 114](#).

The standard installation provides two schema layout options: Siebel Schema without Partitioning and Siebel Schema with Partitioning. One of the Siebel Systems-provided templates listed below is used during the standard installation for the selected layout for your database schema.

Siebel Schema without Partitioning	storage_np.ctf	Contains database storage layout for Siebel Schema with one table per a table space, and one table space per a database. No partitioned table spaces are provided in this storage control file.
Siebel Schema with Partitioning	storage_p.ctf	Contains a partitioning scheme for a set of tables for ASCII; every nonpartitioned table resides in its own segmented table space. Each table space resides in its own database.
	storage_p_e.ctf	Contains a partitioning scheme for a set of tables for EBCDIC; every nonpartitioned table resides in its own segmented table space. Each table space resides in its own database.

See [“About the Siebel Storage Control File Templates” on page 44](#) for descriptions of the templates.

NOTE: If you choose the Siebel partitioned layout, a select subset of Siebel tables is defined as partitioned with predefined partitioning indexes. For a discussion of partitioned table spaces and partitioning indexes, see [“About Siebel Partitioning” on page 55](#).

During the standard installation process, you indicate how you want to execute DDL that was created based on your layout choices. You can execute DDL directly against the database by choosing Run DDL Automatically. Alternatively, by choosing Generate DDL Into Files, you can assign DDL to an output file for later transfer to the DB2 host (using Siebel-provided scripts, your own FTP, or another file transport program) where the DBA applies the DDL using customary tools.

Performing a Standard Installation Under Windows

Perform the following steps for a standard installation under Windows.

To perform a standard installation under Windows

- 1 From the Start menu, navigate to Programs > Siebel Enterprise Server Configuration 7.x.x > Database Server configuration.
The Siebel Gateway Name Server Address screen appears.
- 2 Enter the values for the Siebel Gateway Name Server Address and Enterprise Server Name as you recorded them in your copy of [Appendix A, “Deployment Planning Worksheet.”](#)
 - **Gateway Name Server Address.** The alias of the host on which you installed the Siebel Gateway Name Server.

- **Enterprise Server Name.** The name you gave to your Enterprise Server, for example, siebel.

To continue, click Next.

CAUTION: If you subsequently add a new language (after you complete your initial installation with your initial set of languages), leave the Gateway Name Server Address and Enterprise Server Name fields blank.

- 3 On the Siebel Server Directory screen, perform one of the following tasks:

- Accept the default value displayed in the Siebel Server Directory field. (This is the *SIEBSRVR_ROOT* directory, for example, C: \sea7xx\si ebsrvr.)
- Click Browse to select an alternate directory path.

To continue, click Next.

- 4 On the Siebel Database Server Directory screen, perform either of the following tasks:

- Accept the default path displayed in the Siebel Database Server Directory field. (This is the *DBSRVR_ROOT* directory, for example, C: \sea7xx\dbsrvr.)
- Click Browse to select an alternate directory.

To continue, click Next.

- 5 On the RDBMS Platform screen, select the RDBMS platform you use, in this case, DB2 UDB for z/OS and OS/390.

To continue, click Next.

- 6 On the Siebel Database Operation screen, select Install Database.

To continue, click Next.

- 7 On the Select Installation Operation screen:

- If you are configuring Siebel Database Server software for the first time, click Install Siebel Database.
- If you have already installed your Siebel Database Server software in its base or primary language, and want to install a new language, proceed to ["Installing Multilingual Seed Data" on page 132](#).
- Install Siebel Data Warehouse Database. For information on this installation option, see *Siebel Data Warehouse Installation and Administration Guide*.

To continue, click Next.

- 8 On the ODBC Data Source Name screen, indicate the following values for your ODBC data source:

- Accept the default name of the ODBC data source, SiebSrvr_Siebel, or enter the database alias you prefer to use for the data source.

The database alias must have been previously registered within DB2 Connect, either by using the DB2 Client Configuration assistant or by natively registering to the DB2 Connect catalog, the Database, Node, and DCS entries.

NOTE: If you define database aliases, you must add the TXNISOLOCATION parameter to the database alias entry in the db2cli.ini file and set it to have a value of 1. For information on setting values in the db2cli.ini file, refer to the relevant IBM documentation.

If you want to work with information on a different database server than the one currently pointed to, you can type in the ODBC name of that database server, as long as you previously registered it as an ODBC data source within DB2 Connect.

The Siebel Server installation process automatically creates the data source, using the format SiebSrvr_EnterpriseServerName.

NOTE: There can only be one database server in a Siebel Enterprise Server, but another ODBC data source name could be used in generating a new Siebel Schema DDL.

- Enter the name of the source database DB2 subsystem.

To continue, click Next.

9 On the Database User Name screen, indicate the following values for your database:

- **Database User Name.** Type the ID (for example, SADMIN) used to log into the Siebel Database Server. This user ID should be part of the security group (secondary authorization group) with authorization to set the current SQLID to the schema name.
- **Database Password.** Type the password for the ID used to log into the Siebel Database Server.
- **Database Password (confirm).** Retype the password to confirm it.

To continue, click Next.

10 On the Siebel Schema Qualifier screen, indicate the following values for your Siebel Schema:

- **Siebel Schema Qualifier.** Type the up to eight-character ID that identifies the Siebel Schema owner, for example, SIEBTO.

NOTE: This ID must be in uppercase, must start with a letter, and cannot include any special characters. This is the SQL Authorization ID stored in the CURRENT SQLID special register.

- **Security Group ID / Grantee.** Type the group ID to which schema access is being granted, for example, SSEROLE.

For more information about the group authorization ID privileges, see [Chapter 3, "Security Concepts for z/OS and OS/390."](#)

To continue, click Next.

11 On the Select Installation Type screen, select the appropriate type of installation; for example, to perform a standard installation, select Standard Install.

- **Standard Install.** The standard installation option installs the database, using one of the Siebel-recommended default storage layouts.
- **Customized Install.** For a customized installation, you will need to provide a storage control file, which you created and validated prior to performing this installation. This will allow you to create a custom database layout, best suited to your Siebel deployment. See [“Performing a Custom Installation” on page 114](#) for a description of the custom installation process.

To continue, click Next.

NOTE: The following steps appear after you select Standard Install.

- 12 On the Database Encoding screen, indicate whether your DB2 subsystem is ASCII or EBCDIC. For more information about choosing the code page for your subsystem, see [Chapter 4, “Preparing for Implementation on the DB2 Host.”](#)

- 13 In the Siebel Schema Layout screen, select one of the following layout options:

- **Siebel Schema without Partitioning.** Choose this option if you want all tables only in segmented table spaces.
- **Siebel Schema with Partitioning.** This layout includes a set of tables that is recommended for partitioning. The remaining nonpartitioned tables are in segmented table spaces. For more information on this option, see [“About Configuring Siebel Templates for Partitioning” on page 57](#).

For details about layout considerations, see [Chapter 5, “Configuring the Database Layout.”](#)

To continue, click Next.

- 14 On the Storage Group for Table Spaces screen, indicate the values for the following parameters:

- **Storage Group for Table Spaces.** Type the name for the table storage group.
- **Storage Group for Indexes.** Type the name for the index storage group.

To continue, click Next.

- 15 On the 4KB Buffer Pool Name screen, indicate the values for the following parameters:

- **4KB Buffer Pool Name.** Type the 4-KB buffer pool name for your table spaces or accept the default, BP1. This buffer pool should already be activated and have access to it granted by the DBA.
- **16KB Buffer Pool Name.** Type the 16-KB buffer pool name for your table spaces or accept the default, BP16K1. This buffer pool should already be activated and have access to it granted by the DBA.

For more information about layout considerations, see [Chapter 5, “Configuring the Database Layout.”](#)

To continue, click Next.

- 16 On the 32KB Buffer Pool Name screen, indicate the values for the following parameters:

- **32KB Buffer Pool Name.** Type the 32-KB buffer pool name for your table spaces, or accept the default, BP32K1. This buffer pool should already be activated and have access to it granted by the DBA.

- **Index Buffer Pool Name.** Type the buffer pool name for the indexes, or accept the default, BP2. This buffer pool should already be activated and have access to it granted by the DBA.

To continue, click Next.

- 17** On the Database Name Prefix screen, type the prefix for your database name. The default is SIDB.

This prefix consists of up to four of the first characters in the names of your logical Siebel Databases. This prefix must start with a letter and cannot contain any special characters. All database names end in numbers.

To continue, click Next.

- 18** On the Select Installation Mechanism screen, indicate which installation mechanism you want to use:

- **Generate DDL Into Files.** Select this to generate the DDL required to create the Siebel Schema.

Use Siebel-provided scripts, your own FTP, or a similar file transfer program to transport the DDL (schema.sql and ddlview.sql) to the DB2 host, where the DBA executes it, using customary methods, for example, SPUFI or DSNTPE2.T

To continue, click Next and proceed to [Step 19](#).

- **Run DDL Automatically.** Select this to apply the DDL and DML required to create the Siebel Schema directly against the database, using ODBC to connect.

To continue, click Next and proceed to [Step 21](#).

- 19** On the DDL Commit Frequency screen, choose the number of DDL statements between issuing COMMITs and click Next.

- 20** On the Output Directory screen, indicate the output directory.

Accept the default value, *DBSRVR_ROOT\DB2390\DBOUTPUT\INSTALL*, or enter your new value.

This is the directory that contains schema.sql, ddlview.sql, and the new files to apply DDL on the mainframe.

- 21** On the Log Output Directory screen, indicate the directory where you want the log files to be generated.

Accept the default value, *install_mf*, or enter your new value.

- 22** On the runnow message box, choose from one of the following options:

- To apply the configuration now, press OK.
- To apply the configuration later, press Cancel.

The command line command to apply the configuration later is
C:\sea7x\siebsrvr\bin\siebugp.exe /m master_install_mf.ucf

NOTE: This command is generated dynamically and can vary based on the location of your Siebel Server installation.

23 On the Configuration Parameter Review screen, review the configuration values you entered on the previous Configuration Utility screens:

- To go back and make changes, click Previous to back out until you reach the screen with the parameter you need to change.
- To accept the values you input with no changes, click Finish.

This step completes the installation option to Run DDL Automatically.

If a program or system error occurs and you need to rerun the Siebel Upgrade Wizard, you can do so, starting from the command line at the point at which the wizard failed, by navigating to \siebsrvr\bin and entering:

```
siebugg.exe /m master_install_mf.ucf
```

24 The resulting message box states that the configuration was applied successfully.

To continue click OK.

25 Transfer schema.sql and ddlview.sql to OS/390 and have your DBA apply them to create the Siebel Schema.

For information on how the DBA applies the DDL on the DB2 host, see [“Applying DDL on the DB2 Host” on page 95](#).

26 If you performed a manual install, validate that all tables and indexes are present in the logical schema. For information, refer to the information in [“Preparing the Database Layout” on page 101](#).

When you have successfully completed the step of populating the seed data, you are ready to import the Siebel Repository.

TIP: To preserve a record of the values you entered, locate and print the file named master_install_mf.ucf in the binary subdirectory of your Siebel Server installation directory.

The following files are generated by the installation into the default output directory (DBSRVR_ROOT\DB2390\BOUTPUT\INSTALL) or into the output directory that you designated in [Step 20 on page 89](#):

- **instftp.bat.** Use this batch file to run the Siebel-provided scripts, your own FTP, or a similar file transfer program to transport the DDL (schema.sql and ddlview.sql) to DB2.
- **instftp.txt.** Use this file to transfer files to the mainframe.
- **jobinstl.txt.** Contains JCL and REXX execs to execute the DDL.

Performing a Standard Installation Under UNIX

Perform the following steps for a standard installation under UNIX.

To perform a standard installation under UNIX

- 1 Navigate to `$SIEBEL_HOME` and source environment variables, using one of the following commands, depending on the type of shell you use:

Korn or Bourne shell

```
. ./siebenv.sh
```

TIP: Make sure there is a space between the initial period and `./siebenv.sh`.

C shell

```
source siebenv.csh
```

where:

`$SIEBEL_HOME` is the location of the Siebel Server root directory.

- 2 Verify whether the values `$SIEBEL_ROOT` and `LANGUAGE` are correctly set or are empty by entering:

```
env
```

If the variables are incorrect or missing, reset them using one of the following commands, as appropriate to the shell you use:

Korn or Bourne shell

```
SIEBEL_ROOT=New Value ; export SIEBEL_ROOT
```

```
LANGUAGE=New Value; export LANGUAGE
```

C shell

```
setenv SIEBEL_ROOT New Value
```

```
setenv LANGUAGE New Value
```

- 3 Go to `$SIEBEL_ROOT/bin` and enter:

```
dbsrvr_config.ksh
```

This launches the Database Server configuration wizard.

- 4 Review the values of the following environment variables and confirm whether the settings are correct by entering either Y or N.

`SIEBEL_ROOT` = path of your Siebel installation directory

`LANGUAGE` = language in which you are installing

NOTE: If either the `SIEBEL_ROOT` or `LANGUAGE` value is not set or is incorrect, you must correct them before proceeding.

- 5 Verify and accept the default for the path of your Siebel Server root directory and press ENTER.
- 6 Verify and accept the default for the path of your database server root directory and press ENTER.
- 7 Select your RDBMS platform, in this case, DB2 for z/OS and OS/390.

- 8 From the Siebel Database Operations prompt, select Install Database.
- 9 From the Installation Operations prompt, select Install Database.
 - For installation of an additional database language (2), see [“Installing Multilingual Seed Data” on page 132](#).
 - For installation instructions for Siebel Data Warehouse Database (3), see *Siebel Data Warehouse Installation and Administration Guide*.
- 10 Enter the number corresponding to the language in which you want to run your database. (This is known as your base language.) If you are installing your first language, this is (1).
- 11 On the ODBC Data Source Name screen, accept the default name of the ODBC data source, `si ebSrvr_si ebel`, or enter the database alias you prefer to use for the data source. This alias should have been previously registered within DB2 Connect, either by using the DB2 Client Configuration assistant or by natively registering to the DB2 Connect catalog, the Database, Node, and DCS entries.

NOTE: If you define database aliases, you must add the `TXNI SOLATI ON` parameter to the database alias entry in the `db2cli.ini` file, and set it to have a value of 1. For information on setting values in the `db2cli.ini` file, refer to the relevant IBM documentation.

If you want to work with information on a different database server than the one currently pointed to, you can type in the ODBC name of that database server, as long as you previously registered it as an ODBC data source within DB2 Connect.

The Siebel Server installation process automatically creates the data source, using the format `Si ebSrvr_Enterpri seServerName`.

- 12 Specify the Database User Name.

This is the ID allowed to log into the Siebel Database. This user ID should have authorization to set the `CURRENT SQLID`, for example, `SADMIN`.
- 13 Enter the database password for the Database User Name from [Step 12](#).
- 14 Indicate the following values:
 - Specify the Siebel Schema Qualifier, for example, `SIEBTO`.

This is the up to eight-character (all uppercase) identifier that designates the Siebel Schema. This is also an authorization ID. The Schema Qualifier must be in uppercase, must start with a letter, and cannot contain special characters.
 - Specify the Security Group ID / Grantee or accept the default (`SSEROLE`) by pressing `ENTER`.

This is the group to whom schema access is being granted.
- 15 Specify the type of installation you want to perform (in this case, 1):
 - 1 - Standard Instal l . The standard installation option installs the database using one of the Siebel-recommended default storage layouts.
 - 2 - Customi zed Instal l . For a customized installation, you will need to provide a storage control file, which you created and validated prior to performing this installation. This will allow you to create a custom database layout, best suited to your Siebel deployment. See [“Performing a Custom Installation” on page 114](#) for a description of the custom installation process.

- 16** Enter the number corresponding to the mechanism you want to use for installation:
- 1 - Generate DDL Into Files. Generates DDL required to create the Siebel Schema. Proceed to [Step 17](#).
- You must later use Siebel-provided scripts, your own FTP, or a similar file transfer program to transport the DDL (schema.sql and ddlview.sql) to the DB2 host, where the DBA executes it, using customary methods, for example, SPUFI or DSNTEP2.
- 2 - Run DDL automatically. Select this to apply the DDL and DML required to create the Siebel Schema directly against the database. The Siebel application uses ODBC to apply the required DDL to the DB2 host. Proceed to [Step 20](#).
- 17** On the DDL Commit Frequency screen, choose the number of DDL statements between issuing COMMITs.
- 18** Enter the path to the output directory for the DDL files that you will transfer to your zSeries machine, or accept the default directory:
- ```
dbsrvr/db2390/dboutput/install
```
- NOTE:** If you plan to use a different output directory to hold the DDL than the Siebel default, you must create it before selecting this option.
- 19** Enter the number that corresponds to the code page encoding scheme for your Siebel Database:
- 1- ASCII
  - 2- EBCDIC
- For more information on choosing the database code page, see ["Choosing the Database CCSID" on page 35](#).
- 20** Enter the number corresponding to the Siebel Schema Layout you prefer:
- 1 - Siebel Schema without Partitioning
  - 2 - Siebel Schema with Partitioning
- For details about layout considerations, see [Chapter 5, "Configuring the Database Layout."](#)
- 21** Enter the name for the storage group for table spaces, or accept the default name, SYSDEFLT.
- The name used should be discussed with the DBA before you start the installation.
- 22** Enter the name for the storage group for indexes, or accept the default name, SYSDEFLT.
- 23** Enter the name for your 4-KB buffer pool, or accept the default BP1.
- 24** Enter the name for your 16-KB buffer pool, or accept the default, BP16K1.
- 25** Enter the name for your 32-KB buffer pool, or accept the default, BP32K1.
- 26** Enter the name for your index buffer pool, or accept the default, BP2.
- NOTE:** These buffer pools should be activated and granted by the DBA prior to the installation.

**27** Enter the Database Name Prefix. The default is SI DB.

This prefix consists of up to four of the first characters in the names of your logical Siebel Databases. This prefix must start with a letter and cannot contain any special characters.

**28** Indicate the directory where you want the log files to be generated.

**29** Review the values for accuracy.

- If the values are correct, launch the Siebel Upgrade Wizard (srvrupgwi z) by entering Y.
- To change your previous values enter N, and rerun the DB Server Configuration wizard by entering dbsrvr\_conf i g. ksh. This allows you to reconfigure with different values.

This step completes the installation option to Run DDL Automatically.

If in [Step 16 on page 93](#) you chose installation option, 1 - Generate DDL Into Files, the utility displays the following message after it generates the schema.sql file that contains the DDL to create the Siebel Schema:

Pause #1: Please create Siebel Schema using schema.sql and ddlview.sql located in the Output Directory. Once the objects are created, please select Yes. To stop now and continue installation later please select No. (To resume at a later time, please start the Upgrade Wizard from the command line using option /m master\_install\_mf.ucf).

The DDL is not executed at this time; it is simply generated to a file. (A message indicates that DDL is being executed; this message refers to execution of the generation of DDL statements and the process of writing the statements to a file.)

**NOTE:** If you need to rerun the Upgrade Wizard, you can do so from the command line by navigating to `$SIEBEL_ROOT/bin` and entering:  
`srvrupgwi z /m master_install_mf.ucf`. It restarts where you left off.

**30** Indicate the following:

- Click No to discontinue Siebel Schema (schema.sql) and view (ddlview.sql) generation.

When you click No, the Upgrade Wizard quits.

- To continue, click Yes.

**NOTE:** When Siebel Schema generation is restarted, you will encounter errors during the installation from the restarted point. Disregard these errors and continue with the Siebel Schema creation.

**31** Transfer schema.sql and ddlview.sql to OS/390 and have your DBA apply them to create the Siebel Schema.

For information on how the DBA applies the DDL on the DB2 host, see [“Applying DDL on the DB2 Host” on page 95](#).

**32** If you performed a manual install, validate that all tables and indexes are present in the logical schema. For information, refer to the information in [“Preparing the Database Layout” on page 101](#).

**33** Indicate which action you want to take:

Pause #2: Please ensure that the `schemvld.sql` is empty (which means that Siebel Schema was created successfully). If so, please continue with the UDFs deployment: ftp the files to install UDF functions. (To resume, please start the Upgrade Wizard from the command line using the option `/m master_install_mf.ucf`).

- Only if your DBA has already created the Siebel Schema, click Yes to populate it. (The execution of the DML does not require DBA authority.)

**CAUTION:** When you click Yes, it means you proceed to populate the schema with seed data, and do other manipulations on Siebel tables. The operation will fail if the Siebel Schema has not yet been created, because there is no schema to populate.

- If you want to quit the DB Server Utility until your DBA has had an opportunity to create the Siebel Schema, click No. You can relaunch the wizard at any time to continue with this step.

When you have successfully completed the step of populating the seed data, you are ready to import the Siebel Repository.

**TIP:** To preserve a record of the values you entered, locate and print the file named `master_install_mf.ucf` in the binary subdirectory of your Siebel Server installation directory.

The following files are generated by the installation into the default output directory (`dbsrvr/db2390/dboutput/install`) or into the output directory that you designated in [Step 18 on page 93](#).

- **instftp.bat.** Use this batch file to run the Siebel-provided scripts, your own FTP, or a similar file transfer program to transport the DDL (`schema.sql` and `ddlview.sql`) to DB2.
- **instftp.txt.** Use this file to transfer files to the mainframe.
- **jobinstl.txt.** Contains JCL and REXX execs to execute the DDL.

## Applying DDL on the DB2 Host

When you select the option Generate DDL Into Files, the files `schema.sql` and `ddlview.sql` are generated in the DDL output directory that you specified during the installation. Before you continue with the installation, perform the steps below to apply the DDL to the DB2 host.

### Transferring Files to the DB2 Host

The following instructions are for transferring certain files to the DB2 host.

#### *To transfer files to the DB2 host*

- 1 Navigate to the output directory `DBSRVR_ROOT\DB2390\DBOUTPUT\INSTALL` (Windows) or `dbsrvr_root/db2390/dboutput/install` (UNIX) and open the `instftp.txt` file.
- 2 Edit the `instftp.txt` file using the following information:
  - a Change `&IP` into the IP address or domain name of your DB2 host, for example ZM01.
  - b Change `&Username` into the your own user name, for example SADMIN.

- C** Change *SIEBELQ1* to your own high-level qualifier (HLQ), for example, SADMIN, using a command like Replace All. Save the file.

Following is an example of the instftp.txt file.

```
open &IP
user &Username

quote site cylinders primary=1 secondary=1
quote site recfm=fb lrecl=80 blksize=0

send C:\sea7x\dbsrvr\db2390\dboutput\install\jobinstl.txt
'HLQ.SIEBEL.INST.JOBINSTL'
send C:\sea7x\dbsrvr\db2390\dboutput\install\ddlview.sql
'HLQ.SIEBEL.INST.DDLVIEW'

quote site cylinders primary=10 secondary=25
send C:\sea7x\dbsrvr\db2390\dboutput\install\schema.sql 'HLQ.SIEBEL.INST.SCHEMA'

quit
```

- 3** Double-click instftp.bat (Windows) or issue the following command (UNIX):  

```
ftp -vn < instftp.txt
```
- 4** Enter the password associated with the user name you entered in the instftp.txt file in [Step 2](#).  
 Press any key to continue.
- 5** The log file instftp.log is created in your *DBSRVR\_ROOT\DB2390\DBOUTPUT\INSTALL* (Windows) or *dbsrvr\_root/db2390/dboutput/install* (UNIX) directory.

The log contains information on the file transfer, for example:

Connected to ZM01.siebel.com.

```
220-FTPD1 IBM FTP CS V1R4 at LOOPBACK, 22:44:26 on 2004-02-27.
220 Connection will close if idle for more than 5 minutes.
331 Send password please.
230 SADMIN is logged on. Working directory is SADMIN.
200 SITE command was accepted
200 SITE command was accepted
200 Port request OK.
125 Storing data set SADMIN.SIEBEL.INST.JOBINSTL
250 Transfer completed successfully.
ftp: 45038 bytes sent in 0.00Seconds 45038000.00Kbytes/sec.

200 Port request OK.
125 Storing data set SADMIN.SIEBEL.INST.DDLVIEW
250 Transfer completed successfully.
ftp: 1850 bytes sent in 0.00Seconds 1850000.00Kbytes/sec.

200 SITE command was accepted
200 Port request OK.
125 Storing data set SADMIN.SIEBEL.INST.SCHEMA
250 Transfer completed successfully.
ftp: 8369624 bytes sent in 2.42Seconds 3452.82Kbytes/sec.
```



221 Qui t command recei ved. Goodbye.

- 6 After the transfer is successful, log on to the mainframe, and navigate to the *HLQ.SIEBEL.INST.JOBINSTL* file, for example, *SIEBLQ1.SIEBEL.INST.JOBINSTL*.
- 7 The DSLIST screen reflects the data sets matching the user ID entered previously.
- 8 Edit the *HLQ.SIEBEL.INST.JOBINSTL* data set by entering e to the left of this data set and press ENTER.

The Edit Entry Panel appears.

- 9 As reflected in the comments section change all occurrences of *SIEBELQ1* to your *HLQ* using the following command:

c SIEBELQ1 *HLQ* ALL

**NOTE:** Make sure that your entry is in capital letters.

- 10 Enter submit on the Command line and press ENTER to submit the job.  
This job will create a new data set named *HLQ.SIEBEL.INST.EXEC*.
- 11 On the DSLIST screen, edit the *HLQ.SIEBEL.INST.EXEC* data set by entering e to the left of the data set and press ENTER.
- 12 Enter exec to the left of the SBLINST member to start the installation.
- 13 On the Install Main Menu screen, you will make a selection from the following options:
  1. Generate Siebel DDL - Generate JCL and DDL
  2. Generate View DDL - Generate VIEW DDL
 Enter 1 on the Command line to create JCL and DDL.
- 14 Enter the file name for the DDL then press ENTER to generate the schema.  
For example, enter *HLQ.SIEBEL.INST.SCHEMA* to specify the DDL input data set.  
An edit list will appear for an SQL data set named *HLQ.SIEBEL.INST.SCHEMA.SCH*.
- 15 Press F3 to exit after the job completes.
- 16 Enter e to the left of the *HLQ.SIEBEL.INST.SCHEMA.SCH* data set and press ENTER.
- 17 In the *HLQ.SIEBEL.INST.SCHEMA.SCH* data set, you will edit the following JCL files, in this order:
  - DBJCL
  - TSJCL
  - TBJCL
  - GRJCL
 First, edit the DBJCL JCL file.
  - a Enter e to the left of the DBJCL JCL file and press ENTER.
  - b Edit job card parameters in each file, for example:

```
1000001 //D2PUNLO JOB ACCNT#, CLASS=S, MSGCLASS=X, MSGLEVEL=(1, 1),
000002 // TIME=1440, REGION=OM, NOTIFY=&SYSUID, LINES=9999
000003 //* TYPRUN=SCAN
```

- c** Enter an asterisk (\*) to the left of the TYPRUN parameter.
- d** Enter submit on the Command line and press ENTER to submit the job.

Make the same edits in the remaining files.

After each job is completed, verify that the return code is 0 (zero) to make sure that no errors have occurred and that the job successfully completed.

- 18** Go to [Step 11 on page 97](#) and follow the instruction to edit the *HLQ.SIEBEL.INST.EXEC* file to generate the View DDL file.

- 19** Select 2 on the Install Main Menu screen to generate the View DDL file.

- 20** Enter the DDL file name, for example, *HLQ.SIEBEL.INST.DDLVIEW* to specify the DDL input data set.

Press ENTER to generate a DDLVIEW SQL data set named *HLQ.SIEBEL.INST.DDLVIEW.VIE*.

- 21** Enter e to the left of the *HLQ.SIEBEL.INST.DDLVIEW.VIE* data set and press ENTER.

- 22** In the *HLQ.SIEBEL.INST.DDLVIEW.VIE* data set, you will edit the VIEWJCL file.

- a** Enter e to the left of the VIEWJCL JCL file and press ENTER.
- b** Edit job card parameters in each file, for example:

```
1000001 //D2PUNLO JOB ACCNT#, CLASS=S, MSGCLASS=X, MSGLEVEL=(1, 1),
000002 // TIME=1440, REGION=OM, NOTIFY=&SYSUID, LINES=9999
000003 //* TYPRUN=SCAN
```

- c** Enter an asterisk (\*) to the left of the TYPRUN parameter.
- d** Enter submit on the Command line and press ENTER to submit the job.

After each job is completed, verify that the return code is 0 (zero) to make sure that no errors have occurred and that the job successfully completed.

- 23** Return to the mid-tier to continue the database server installation.

Click Yes on the following screen after all the processes are complete:

Pause #1: Please create Siebel Schema using schema.sql and ddlview.sql located in the Output Directory. Once the objects are created, please select Yes. To stop now and continue installation later please select No. (To resume at a later time, please start the Upgrade Wizard from the command line using option /m master\_install\_mf.ucf).

- 24** Make sure that the schemvld.sql file, generated during the installation and placed in the default output directory (*DBSRVR\_ROOT\DB2390\DBOUTPUT\INSTALL* on Windows or *dbsrvr\_root/db2390/dboutput/install* on UNIX) or into the output directory that you designated, is empty.

Click Yes on the following screen to complete the manual installation:

Pause #2: Please ensure that the `schemvld.sql` is empty (which means that Siebel Schema was created successfully). If so, please continue with the UDFs deployment: ftp the files to install UDF functions. (To resume, please start the Upgrade Wizard from the command line using the option `/m master_install_mf.ucf`).

**NOTE:** If you would like to save the `schema.sql` file with the complete Siebel Schema DDL, which was generated for the entire Siebel Schema, rename `schema.sql` after you apply `schema.sql` on the mainframe.

## Validating Siebel Schema Creation

The second pause generated the `schemvld.sql` and `schema.sql` files. The `schemvld.sql` file contains validation sql, based on the newly created Siebel Schema. The `schema.sql` file contains SQL statements for missing Siebel schema objects, if such objects are found. Normally, `schema.sql` is empty.

When reviewing the `schema.sql` file, two possible options exist:

- The file is empty, which means the entire schema was created successfully.
- The file is not empty, which means the schema was not created successfully.

This is likely a result of the DBA missing a step during the manual schema creation part on the DB2 host. The file contains the missing objects of the schema (the *Delta* schema). The DBA needs to apply the file with the missing object to complete a full schema creation.

**NOTE:** If you are creating a Siebel Schema with partitioning, the generated `schema.sql` file contains drop and create triggers which are acceptable and can be ignored.

## Deploying User Defined Functions

There are two user defined functions (UDFs) that are provided in the Siebel application: `nextseq` and `exrate`. The `nextseq()` UDF generates sequential values for EIM Export and the `exrate()` UDF is used for currency aggregation functions. After the schema validation, UDFs must be installed. Following are the instructions on installing the UDFs during an installation.

**NOTE:** You must have Workload Manager (WLM) installed and configured to run the `nextseq` and `exrate` user-defined functions.

### To deploy user defined functions

- 1 Navigate to the default installation output directory and modify the `udfftp.txt` FTP file:
  - a Change the IP address to the IP or domain name of your mainframe.
  - b Change the user ID to a valid TSO user ID.
  - c Change the `SIEBELQ1` high level DSN qualifier to a valid DSN qualifier.
- 2 Double-click `DBSRVR_ROOT\DB2390\DBOUTPUT\INSTALL\udfftp.bat` to execute the batch FTP script to send files to the mainframe (Windows), or issue the following command (UNIX):

```
ftp -vn < udfftp.txt
```

The script sends the following files to the mainframe from the *DBSRVR\_ROOT\DB2390\STORPROC\ZOS* directory (Windows) or the *\$DBSRVR\_ROOT/DB2390/STORPROC/ZOS* directory (UNIX):

- DB2390/STORPROC/ZOS/SIEBEL.UDF.RECEIVE.JCL
- DB2390/STORPROC/ZOS/SIEBEL.UDFLOAD.XMIT
- DB2390/STORPROC/ZOS/SIEBEL.UDF.CNTL.XMIT
- DB2390/STORPROC/ZOS/SIEBEL.UDFDDL.XMIT
- DB2390/STORPROC/ZOS/SIEBEL.UDFDBRM.XMIT

- 3 Modify the *USER\_ID.SIEBEL.UDF.RECEIVE.JCL* dataset according to instructions provided in this JCL file and submit it.

- 4 After the jobs complete, run them in the UDF.CNTL PDS.

Make sure that the condition equals 0.

## Setting Up the SQL Delimiter on DB2

Because the file *schema.sql* uses a slash as a delimiter, you must set up the slash as the delimiter in the tool (SPUFI or DSNTEP2) you are using to apply the DDL.

Siebel eBusiness Applications ships files with slashes for delimiters, because *schema.sql* might contain triggers for table partitioning that use semicolons. For example:

```
CREATE TRIGGER Q202999.PTH0393 NO CASCADE BEFORE INSERT ON
Q202999.S_ACT_EMP REFERENCING NEW AS N FOR EACH ROW MODE DB2SQL
WHEN (N.EMP_ID IS NOT NULL) BEGIN ATOMIC SET N.PARTITION_COLUMN =
RIGHT(N.EMP_ID, 2); END
/
```

**NOTE:** If you are not using table space partitioning, it is safe to replace slashes with semicolons.

### DSNTEP2

If you are going to run the DDL in batch mode, using DSNTEP2, use the parameter *SQLTERM(/)* to set this. The example below illustrates how this appears in the *SYSTSN DD* statement.

**NOTE:** In the following example, *Q202* is the subsystem name.

DSNTEP2 example:

```
//SYSTSN DD *
DSN SYSTEM(Q202)

RUN PROGRAM(DSNTEP2) PLAN(DSNTEP71) PARMS(' /SQLTERM(/)') -
 LIB(' DSN710.RUNLIB.LOAD')
```

END

## SPUFI

If you will use SPUFI, perform the following procedure.

### *To set the delimiter in SPUFI*

- Set CHANGE DEFAULTS to YES.

When the CURRENT SPUFI DEFAULTS panel appears, change the SQL TERMINATOR to a slash mark.

**NOTE:** If triggers are not being used, you can change the slashes back to semicolons. If using the TSO editor, perform the following procedure.

### *To replace slashes with semicolons in the TSO editor*

- Enter the following command:

```
CHANGE X' 61' X' 5E' ALL
```

If your platform at midtier is Microsoft Windows, return to [Step 25 on page 90](#); if your platform is UNIX, [Step 31 on page 94](#).

## Preparing the Database Layout

When you select the standard installation method, the DB Server Configuration wizard creates a storage control file (called storage.ctl) based on the information that you provide in response to prompts.

When you select the custom installation method, you must create a storage control file before you perform Schema installation. There are several methods you can use to create a storage control file:

- **Use an existing storage control file.** You can specify the name of an existing storage control file in a custom installation.
- **Modify one of the Siebel-provided storage control file templates.** To modify a template, you can edit the file directly, or you can use a Siebel tool called the Database Storage Configurator (dbconf.xls). For more information, see [“About Creating or Modifying the Database Layout” on page 52](#).
- **Extract a new storage control file for custom installation.** To create a storage control file based on the layout of the existing Siebel Schema, use the DB Server Configuration wizard - Extract Utility. This utility allows you to create the storage control file by “reverse-engineering” the DB2 catalog. For more information, see [“Extracting a Storage Control File from the DB2 Catalog” on page 102](#).

## Extracting a Storage Control File from the DB2 Catalog

DBAs can create a storage control file for use during installation by using another Siebel utility to extract it from the DB2 catalog. Access the extract utility through the DB Server Configuration wizard.

This utility allows you to extract the storage layout of your source database from the DB2 catalog. The utility creates a new storage control file that can be used during the custom installation process. You can use the extract utility at any time.

**NOTE:** You must validate the storage control file after you extract it and after you modify it. You must also perform a validation before you apply the schema.sql on the DB2 host. See “Validating the Storage Control File” on page 105.

## Accessing the Utility Under Windows

Follow the procedure below to access the utility under Microsoft Windows.

### *To extract the storage control file under Windows*

- 1 Launch the Database Server configuration wizard:

Choose Start > Programs > Siebel Enterprise Server Configuration 7.x.x > Database Server configuration.

or

- a Open a command prompt.
- b Go to the `SI EBSRVR_ROOT\bin` directory.
- c Run the following command:

```
ssi ncfgw -I ENU -v Y
```

- d In the screen that appears, select `dbsrvr.scm`.

- 2 In the Gateway Name Server Address screen, type the following values as you recorded them in your copy of [Appendix A, “Deployment Planning Worksheet.”](#)

- **Gateway Server Name Address.** The alias of the host on which you installed the Siebel Gateway Name Server.
- **Enterprise Server Name.** The name you gave to your Enterprise Server, for example, `si ebel`.

To continue, click Next.

- 3 In the Siebel Server Directory screen, use either of the following methods to select your directory:

- Accept the default value displayed in the Siebel Server Directory field. (This is the `SI EBSRVR_ROOT` directory, for example, `C:\sea7xx\si ebsrvr`.)
- Use the browse button to select an alternate directory path.

To continue, click Next.

- 4 In the Siebel Database Server Directory screen, perform either of the following tasks:
  - Accept the default path displayed in the Siebel Database Server Directory field. (This is the *DBSRVR\_ROOT* directory, for example, C:\sea7xx\dbsrvr.)
  - Use the Browse button to select a different database directory.

To continue, click Next.
- 5 In the RDBMS Platform screen, choose IBM DB2 UDB for z/OS and OS/390, and then click Next.
- 6 In the Siebel Database Operation screen, select Run Database Utilities, and then click Next.
- 7 In the Database Utility Selection screen, select Configure Database, and then click Next.
 

The Configure Database screen appears with the following options:

  - Extract Storage Control File.
  - Validate Storage Control File. This will validate that a specified template and a specified schema will work on the DB2 host.
- 8 In the Database Configuration Options screen, select Extract Storage File, and then click Next.
- 9 In the ODBC Data Source Name screen, accept the default name of the ODBC data source after verifying that it is correct (make sure that this data source has the same name as the subsystem qualifier), or enter a different data source name.
 

**NOTE:** The Siebel Server installation process automatically creates the data source, using the format *Si ebSrvr\_Enterpri seServerName*. To find the name of your ODBC data source, navigate to Start > Settings > Control Panel > Administrative Tools > Data Sources (ODBC). Click the System DSN tab and you will find the name of your ODBC data source.
- 10 In the Database User Name screen, type your source database username and password, confirm the password by typing it again, and then click Next.
 

The configuration utility will validate your username and password before it will proceed to the next screen. If you entered a valid username and password, the Schema Qualifier screen appears.
- 11 In the Siebel Schema Qualifier screen, type the schema qualifier and group ID for your existing database, and then click Next.
 

The schema qualifier is also an authorization ID. The schema qualifier must start with a letter and cannot contain special characters.
- 12 In the Security Group ID screen, type the user ID of the group to whom schema access is granted, for example, SSEROLE.
 

To continue, click Next.
- 13 In the Extract Options screen, select Extract from Catalog, and then click Next.
 

This option extracts the storage layout of your source database from the DB2 catalog as-is. The output will be a new storage layout.
- 14 In the Storage File screen, type the name of the new storage control file, for example *DBSRVR\_ROOT\DB2390\my\_storage\_file.ctl*, and then click Next.

- 15** In the Configuration Parameter Review screen, review the parameters that you entered, then click Finish to confirm them.

**NOTE:** Passwords are encrypted and will not appear in plain text either in the user interface or in the upgrade configuration files (UCF files). After a password is entered, it will always appear in encrypted form. If you need to use another password, you must rerun the configuration utility.

- If you need to go back to make changes, click Previous to back out until you reach the screen with the parameter you need to change. Enter the valid parameter, and then click Next until you reach the Configuration Parameter Review screen again.
- To accept the values you input with no changes, click Finish. A message box gives you the choice of applying the configuration now or applying it later. This screen also provides the option you will need in order to apply the configuration later from the command line.

- 16** When the runnow screen appears, click OK to apply the configuration now.

- 17** When the Siebel Upgrade Wizard launches, click OK to perform the migration.

The Siebel Upgrade Wizard creates the storage control file.

## Accessing the Utility Under UNIX

Follow the procedure below to access the utility under UNIX.

### To create the storage control file under UNIX

- 1** Source environment variables from `$SIEBEL_ROOT`.

`SIEBEL_ROOT` should be the path of your Siebel installation directory.

`LANGUAGE` should be set to the language in which the configuration wizard prompts appear, for example, `enu` for U.S. English.

If either of these values is incorrect or empty, reset them using one of the following commands, as appropriate to the shell you use:

#### **csh**

```
setenv SIEBEL_ROOT New Value
```

```
setenv LANGUAGE New Value
```

#### **ksh**

```
SIEBEL_ROOT=New Value ; export SIEBEL_ROOT
```

```
LANGUAGE=New Value; export LANGUAGE
```

- 2** Go to `$SIEBEL_ROOT/bin` and enter:

```
dbsrvr_config.ksh
```

This launches the Database Server configuration wizard.



- 3 Review the values of the following environment variables and confirm whether the settings are correct by entering either Y or N.  
**NOTE:** If either the `SI_EBEL_ROOT` or `LANGUAGE` value is not set or is incorrect, you must correct it before proceeding.
- 4 Specify the path of your Siebel Server root directory, or accept the default by pressing ENTER.
- 5 Specify the path of your database server root directory, or accept the default by pressing ENTER.
- 6 Enter the number that corresponds to your database platform.
- 7 In the Select Database Operation screen, choose Run Database Utilities (5).
- 8 In the Database Utility Selection screen, choose Configure Database (3).
- 9 In the Database Configuration Options screen, choose Extract Storage Control File (1).
- 10 Enter the number that corresponds to the language you are using, typically 1 for English.
- 11 Enter your source ODBC connection name.
- 12 Enter your source database username and password, and then confirm the password by typing it again.
- 13 Enter the schema qualifier in uppercase letters for your existing database.
- 14 Enter the group ID for your existing database, or accept the default (SSEROLE) by pressing ENTER.
- 15 In the Extract Options screen, select Extract from Catalog (1).
- 16 Enter the name of the new storage control file, for example *DBSRVR\_ROOT/DB2390/my\_storage\_file.ctl*
- 17 When the configuration parameters are listed for your review and you are asked if you want to run `srvrupgwi z`, enter Y for yes.
- 18 When the Siebel Upgrade Wizard launches, enter Y to perform the operation.  
The Siebel Upgrade Wizard creates the storage control file.

## Validating the Storage Control File

You must validate the storage control file after you extract it or after you modify it. The validation will produce two output files in the log directory under *SI\_EBSRVR\_ROOT*:

- `db_conf_i g_si_ebel . l og`
- `db_conf_i g_val i dati on. l og`

**NOTE:** When validating the extracted storage control file, the validation job calls for the `schema.ddl` file, as an example. You should make a copy of `ddl.ctl` and name it `schema.ddl`, keeping the copy in the same location as `ddl.ctl`.

### *To validate the storage control file under Windows*

- 1 Run the Siebel Software Configuration Utility as you did in [“Extracting a Storage Control File from the DB2 Catalog” on page 102](#).
- 2 From the Database Utility Selection screen, choose **Configure Database** and click **Next**.  
The **Configure Database** screen appears with the following options:
  - **Extract Storage Control File**
  - **Validate Storage Control File**
- 3 Choose **Validate Storage Control File** and click **Next**.
- 4 On the ODBC Data Source screen, accept the default name of the ODBC data source after verifying that it is correct (make sure that this has the same name as the subsystem qualifier), or enter a different data source name.

**NOTE:** The Siebel Server installation process automatically creates the data source, using the format `Si ebSrvr_Enterpri seServerName`. To find the name of your ODBC data source, navigate to **Start > Settings > Control Panel > ODBC data source**. Click the **System DSN** tab and you will find the name of your ODBC data source.

To continue, click **Next**.

- 5 On the Database User Name screen, indicate the source user name and password for your database:
  - **Database User Name.** Type the ID allowed to log in to the Siebel Database. This user ID should have authorization to set CURRENT SQLID.
  - **Database Password.** Type the password for the ID used to log in to the Siebel database.
  - **Database Password (confirm).** Retype the password to confirm it.

To continue, click **Next**.

The configuration utility will validate your username and password before it will proceed to the next screen. If you entered a valid username and password, the Schema Qualifier screen appears.

- 6 On the Siebel Schema Qualifier screen, type the up to eight-character identifier that designates Siebel Schema for your database. This is also an authorization ID. The schema qualifier must start with a letter and cannot contain special characters.

To continue, click **Next**.

- 7 On the Security Group ID screen, type the user ID of the group to whom schema access is granted, for example, SSEROLE.

To continue, click **Next**.

- 8 On the Schema File screen, type the directory paths for the following files:

- The schema file

If your Siebel Schema has been modified, use `schema.ddl`. For the Siebel out-of-the-box Schema, use `ddl.ctl`.

- The storage control file that you want to validate (for example, storage.ctl)

If schema file verification succeeds, the Configuration Parameter Review screen appears.

- 9 On the Configuration Parameter Review screen, review the configuration values you entered on the previous Configuration Utility screens.

**NOTE:** Passwords are encrypted and will not appear in plain text either in the user interface or in the upgrade configuration files (UCF files). After a password is entered, it will always appear in encrypted form. If you need to use another password, you must rerun the configuration utility.

- If you need to go back to make changes, click Previous to back out until you reach the screen with the parameter you need to change. Enter the valid parameter, and then click Next until you reach the Configuration Parameter Review screen again.
- To accept the values you input with no changes, click Finish.

A message box appears with the prompt:

To apply the configuration now, press "OK".

To apply the configuration later, press "Cancel".

- 10 To validate that storage.ctl works, connect to the catalog and run ddlimp in validation mode.
- 11 Review the log file (ddl imp. log) that is generated in the log subdirectory under *SI EB SRVR\_ROOT*.

### ***To validate the storage control file under UNIX***

- 1 Verify that all servers are stopped:
  - Stop the Siebel Gateway Name Server by typing: stop\_ns
  - Stop the Siebel Servers by typing: stop\_server ALL
- 2 Depending on your shell, source environment variables from *\$SI EBEL\_ROOT* by typing:

#### **Korn shell**

. si ebenv. sh

#### **C shell**

source si ebenv. csh

- 3 Navigate to *\$SI EBEL\_ROOT/bin* and enter:

dbsrvr\_config.ksh

This launches the Database Server configuration wizard.

- 4 Review the values of the following environment variables and confirm whether or not the settings are correct by entering either Y or N.
  - SI EBEL\_ROOT
  - LANGUAGE

**NOTE:** If either the SI EBEL\_ROOT or LANGUAGE value is not set or is incorrect, you must correct it before proceeding.

- 5 Specify the path of your Siebel Server root directory or accept the default by pressing ENTER.
- 6 Specify the path of your database server root directory or accept the default by pressing ENTER.
- 7 Enter the number that corresponds to IBM DB2 UDB for z/OS and OS/390, then press ENTER.
- 8 In the Select Database Operation screen, choose Run Database Utilities (5).
- 9 In the Database Utility Selection screen, choose Configure Database (3).
- 10 In the Database Configuration Options screen, choose Validate Storage Control File (2).
- 11 Enter your source ODBC connection name.  
**NOTE:** This needs to be the same as the actual name of your subsystem. That is, your database alias within DB2 Connect must be the same as the actual name of your DB2 subsystem.
- 12 Enter your source database username and password.
- 13 Enter the schema qualifier for your existing database (enter this value in all uppercase characters).
- 14 Enter the security group ID for your existing database, or accept the default (SSEROLE) by pressing ENTER.
- 15 Specify the storage control file that you would like to validate.  
The configuration parameters are listed.
- 16 When the program asks if you would like to run `srvrupgwi z`, enter Y for Yes.  
The storage control file is validated.
- 17 Review the log file generated by `ddlmp (ddl i mp. l og)` in the log directory under *SI EBSRVR\_ROOT*.

## Using the Siebel Database Storage Configurator

The installation of your Siebel eBusiness Applications includes a Microsoft Excel tool called the Siebel Database Storage Configurator (`dbconf.xls`). This tool facilitates the configuration of your Siebel Schema by allowing you to import and edit the attributes from one of the Siebel-provided storage control file templates. You can also use this tool to configure an existing storage control file.

The file associated with the Siebel Database Storage Configurator is called `dbconf.xls`. This file is installed in the DB2390 subdirectory of your *DBSRVR\_ROOT* directory. In order to open `dbconf.xls`, you must have Microsoft Excel installed on your workstation.

After you have edited a storage control file template using the Database Storage Configurator, save the file with a new name, then direct the DB Server Configuration wizard to your newly created (saved) and configured storage control file during installation (as described in [“Performing a Custom Installation” on page 114](#)). When the DB Server Configuration wizard prompts you to specify a storage control file, select your new storage control file. The output DDL is generated from the information in the storage control file that you select.

## Configuring a Siebel Storage Control File

The following procedure uses the template `storage_p.ctl` as an example to illustrate general use of the Database Storage Configurator.

**TIP:** It is recommended that you preserve the original `dbconf.xls` spreadsheet, and save your modified spreadsheet with a new name. Saving the spreadsheet allows you to review your current database layout while continuing to work on the storage control file; for example, because the spreadsheet allows you to sort data, you can easily identify all table spaces that belong to the same database.

### *To configure one of the Siebel storage control files*

- 1 Open `dbconf.xls` and select Enable Macros when prompted.

**NOTE:** If you are not prompted to enable macros, verify that your Microsoft Excel security setting is Medium. Launch Excel, choose Tools > Macro > Security. Verify that Security Level is set to Medium. If necessary, reset the security level to medium, then restart your computer.

When the spreadsheet opens, the Home tab is active.

- 2 Import the storage control file you want to use as your template:

- a Click Import.
- b Go to the directory where your files with a `.ctl` suffix are located.
- c Double-click the appropriate file.

When the import process is complete, a message appears, stating that you have imported the storage control file successfully.

Click OK.

- 3 When the message, "Please enter default values for your system" appears, either type the values for the following parameters into the corresponding fields, or accept the default values.

**NOTE:** Replace variables that are preceded with "\$" with actual values for your deployment.

- Table Storage Group for Table
- Index Storage for Indexes
- 4K Buffer Pool Name
- 16K Buffer Pool Name
- 32K Buffer Pool Name
- Index Buffer Pool Name
- Database Name Prefix
- Encoding Scheme (either ASCII or EBCDIC)

- 4 Click Set.
- 5 Display the template defaults by clicking the Default tab.

- 6 Review the values and update them as required.

**NOTE:** The default values are the combined values that apply to each individual object represented by the tabs; the values can be changed within each of those tabs.

The parameters in the spreadsheet correspond to native DB2 syntax. For example, the parameters Pri Qty (Primary Quantity) and SecQty (Secondary Quantity) are used during the Siebel schema creation.

As you review the values under other tabs such as Tablespace, an empty parameter cell indicates that the cell takes the default from the value shown in the Default tab value.

When you create a new extension table, the Database Configurator takes the Siebel default database and table space (SIEBTS00) values of the template used unless you explicitly define new values.

- 7 Review the databases by clicking the Database tab.

Be aware that the values visible on any tab also reflect the defaults for any objects at a lower level from the object for which the tab exists. So, if no locksize is shown on for a table space under the Tablespace tab, that means that it takes the default from the database object it belongs to; for example, locksize on the database object level applies to all table spaces created in a given database unless the locksize is specified explicitly in the table space object.

- 8 Review the default table spaces by clicking the Tablespace tab.

To review any partitions, do either one of the following:

- Click twice on the Tablespace tab. (Clicking on the Tablespace tab alternates between the table spaces and table space partition worksheets.)
- Select a row that displays a value in the Partition column greater than zero, then click Show TSPARTs.

This displays the rows of corresponding partitions.

- 9 Edit the values as desired, and then step off the cell to save the value.

- 10 Review the default tables by clicking the Table tab.

- 11 When you have updated the template, click the Home tab and then click Export.

The Export worksheets to storage control file screen appears.

- 12 Rename the file and save it to a location that is easy for you to find during installation.

- 13 To make the Configurator ready for use with a different template, from the Home tab, click Clean.

A text box appears letting you know that you have cleaned all data successfully.

### Validating Your Work

You can use the Validation button at any time to validate the syntax in the spreadsheet. Your worksheets will be more consistent if you wait to validate until after you have finished editing.

***To validate your worksheet***

- 1 Click the Validate button.

The Storage Configurator validates the new value and if the value is incorrect it appears in red. (A text box also appears prompting you to review the invalid settings in red text.)

- 2 Update any incorrect value, step off the cell, and revalidate.

The red highlighting disappears when the validation shows that the value is correct.

**Partitioning Tables and Indexes**

The following two procedures demonstrate how to use the Database Storage Configurator to partition tables after you have worked out a partitioning scheme. For guidelines about partitioning, see [Chapter 6, "Understanding Siebel Partitioning."](#)

**NOTE:** The spreadsheet validation is not a substitute for using the DB Server Configuration wizard to validate the storage control file. The configuration utility option to "Validate Storage Control File" validates the new storage control file against the physical layout of the database.

**Decreasing the Number of Partitions**

This procedure describes how to decrease the number of partitions in a table space.

***To decrease the number of partitions***

- 1 Launch the Database Storage Configurator as described in ["Configuring a Siebel Storage Control File" on page 109](#) and import the .ctl file you want to amend.
- 2 Click the Tablespace tab on the Database Storage Configurator.
- 3 Locate the row with the Tablespace tab in it that you want to change, and change the value in the Partitions column to reflect the number of new partitions you want to create.
- 4 Save the new value by stepping off the cell.
- 5 Click Show TSPARTs.
  - If a message box appears, prompting you that there are additional table space partitions for this table, and requesting that you review and delete them, do so now.
  - If everything is in order, you receive a message indicating that the operation was successful.
- 6 Click the Partitioning Index tab and locate the partitioning index associated with this table space.
- 7 Update the number of partitions to match the number of table space partitions you entered previously, then step off the cell to save the value.
- 8 Click Show Index Parts.
  - If a message box appears, prompting you that there are additional partitioning indexes for this table space, and requesting that you review and delete them, do so now. Proceed to [Step 9](#).
  - If everything is in order, you receive a message indicating that the operation was successful.

- 9 Review the values in the LimitKey column and update them appropriately.

**NOTE:** When using Excel, you must type the first quote as two single quotes. Excel saves it as a single quote when you step off the cell. If you do not enter two single quotes, Excel deletes it entirely when you step off the cell.

- 10 Make any other edits needed to the template you are using and follow the steps under [“Configuring a Siebel Storage Control File” on page 109](#) to verify and save your changes to a location that will be easy to find when you or someone else installs the Siebel Schema.

### Increasing the Number of Partitions

Usually, the partitioning is done for a table. You must adjust the related table space and partitioning index. The following procedure describes how to increase the number of partitions for a table and index in a given table space.

#### *To increase the number of partitions*

- 1 Launch the Database Storage Configurator as described earlier in this chapter under [“Configuring a Siebel Storage Control File” on page 109](#) and select the Table tab.
- 2 Locate the name of the partitioned table space for which you want to increase the number of partitions.
- 3 Click the Tablespace tab and search for the corresponding table space.
- 4 Increase the value in the Partitions column for that table space as appropriate and step off the cell to save it, but keep your cursor in the same row. (If you do not keep your cursor in the same row, you cannot get to the table space partitions by clicking the Show TSPARTs button.)
- 5 Click Show TSPARTs.

When you increase the value in the Partitions column, you receive a message indicating that the partitions were added successfully.

- 6 Click the Partitioning Index tab and locate the partitioning index associated with the table you are operating on.
- 7 Increase the number of partitions to match the value you previously entered for the associated table space, and step off the cell to save the value, but keep your cursor on the same row.
- 8 Click Show Index Parts.

A text box appears prompting you that the extra partitions were created successfully.

- 9 Review the LimitKey values and update them as appropriate.

**NOTE:** When using Excel, you must type the first quote as two single quotes. Excel saves it as a single quote when you step off the cell. If you do not enter two single quotes, Excel deletes it entirely when you step off the cell.

- 10 Make any other edits needed to the template you are using. Then follow the steps under [“Configuring a Siebel Storage Control File” on page 109](#) to verify and save your changes to a location that will be easy to find when you or someone else installs the Siebel Schema.



## Working with CLOBs

Your Siebel application is delivered with the objects needed to create character large objects (CLOBs), even if you did not choose CLOBs as an option. These objects are used to create CLOBs only at the table level to prevent excessive processing overhead.

### To activate a CLOB

- 1 Launch the Database Storage Configurator as described in [“Configuring a Siebel Storage Control File” on page 109](#) and select the Table tab.
- 2 Review the contents of the CLOB column.
- 3 To activate a particular CLOB, type Yes in the appropriate row in the CLOB column, then save the value by stepping off the cell.

**NOTE:** The auxiliary table space name displays a table name that is used as a pointer to a real auxiliary table space. The parameters displayed under this tab are the only parameters allowed to change for the auxiliary object.

## Creating and Executing the DDL

You are ready to transfer the output DDL files (generated from the information in the storage control file that you selected) to the mainframe, where you will execute them.

## Moving DDL Files to the S/390 or zSeries Machine

When you are ready to transfer DDL files for execution, you will use Siebel-provided scripts or FTP (or any other available file transfer option) to move the following files from the output directory (the default output directory is DBSRVR\DB2390\DBOUTPUT\INSTALL on Windows or dbsrvr/db2390/dboutput/install on UNIX) to the S/390 or zSeries machine:

- **ddlsplit.txt.** Contains a REXX routine that can be used to split the DDL into more manageable pieces. This file must be moved to a PDS with RECFM=80 and a LRECL of 80.
- **instddl.txt.** Contains sample JCL to execute the DDL using DSNTEP2. This file must be moved to a PDS with RECFM=FB and LRECL=80.
- **schema.sql.** Contains DDL for all Siebel DB2 objects.
- **Alter statements.** (Optional) Contains the alter statements for the object sizing. This file must be moved manually.

A sample FTP file named instftp.txt is shipped with Siebel eBusiness Applications.

## Splitting the DDL Job

In `HLQ.SIEBEL.INST.SCHEMA.SCH` on your DB2 host there are members to execute the install DDL. You can create the DDL as one job or break it into several jobs:

- SCHDDL. Create all of the DDL objects as one job.
- Create the DDL as several jobs:
  - DB. Create database statements
  - TS. Create table-space statements
  - TB. Create table and index statements
  - GRANT. All grant statements

In the *HLQ.SIEBEL.INST.SCHEMA.SCH* data set, the JCL to execute the DDL as one job is:

```
SCHJCL
```

The JCL to create the DDL as several jobs is as follows. These commands must be run in the order shown:

```
DBJCL
```

```
TSJCL
```

```
TBJCL
```

```
GRJCL
```

## Editing the Member JCL Files

First edit the DBJCL JCL file. After editing the DBJCL JCL file, repeat the following instructions on the remaining JCL files.

### *To edit the member JCL files*

- 1 Enter the letter e to the left of the DBJCL JCL file and press ENTER.
- 2 Edit job card parameters in each file, for example:
 

```
1000001, //D2PUNLO JOB ACCNT#, CLASS=S, MSGCLASS=X, MSGLEVEL=(1, 1),
000002, //TIME=1440, REGION=OM, NOTIFY=&SYSUID, LINES=9999, USER=SBLDBA
000003, //*TYPRUN=SCAN
```

Enter an asterisk (\*) to the left of the TYPRUN parameter.

- 3 Enter submit on the command line to submit the job.
- 4 Repeat these steps on the next JCL file.

After you have edited all the JCL files and submitted the jobs, check the job status to make sure that no errors occurred.

## Performing a Custom Installation

The custom installation option provides significant flexibility for configuring the storage layout for the Siebel application installation.

The custom installation consists of the following steps:

- Prepare a storage control file. (See [“Preparing the Database Layout” on page 101.](#))
- Run the DB Server Configuration wizard, select the Custom Installation option, and provide your customized storage control file (prepared during the previous step) that will be used in the storage configuration for the Siebel Schema. (See the following procedure for your operating system.)
- Populate the seed data. (See the following procedure for your operating system.)
- Import the Siebel Repository. (See [Chapter 9, “Siebel Repository Import, Export, and Postinstallation Tasks.”](#))

## Performing a Custom Installation Under Windows

Perform the following procedure if you are installing Siebel eBusiness Applications on the Windows operating system.

### *To perform a custom installation under Windows*

- 1 Launch the Database Server Configuration wizard and proceed as in the Standard Installation option until the Select Installation Type screen appears ([Step 11 on page 87](#)).
- 2 In the Select Installation Type screen, select Customized Installation.  
To continue, click Next.
- 3 In the Select Installation Mechanism screen, indicate which installation mechanism you want to use:
  - **Generate DDL into Files.** Select this to generate the DDL required to create the Siebel Schema.  
  
**NOTE:** The DDL is not executed at this time; it is simply generated to a file. A message indicates that DDL is being executed; this message refers to execution of the generation of DDL statements and the process of writing the statements to a file.  
  
 You must later use Siebel-provided scripts, your own FTP, or a similar file transfer program to transport the DDL (schema.sql and ddlview.sql) to the DB2 host, where the DBA executes it, using customary methods, for example, SPUFI.  
  
 To continue, click Next and proceed to [Step 4](#).
  - **Run DDL Automatically.** Select this to apply the DDL and DML required to create the Siebel Schema directly against the database.  
  
 The Siebel application uses ODBC to apply all necessary steps in this installation.  
  
 To continue, click Next and proceed to [Step 5 on page 116](#).
- 4 In the DDL Output Directory screen, enter the path to the output directory where you want to deposit the generated DDL for later transport to the DB2 host. The default directory is `SIEBEL_ROOT\DBSRVR\DB2390\DBOUTPUT\INSTALL`.

- 5 In the Storage Control File screen, use the Browse button to indicate the path to the customized storage control file that you want to use to configure storage on the DB2 host.

This can be either a Siebel-provided template or a storage control file that you have already customized using any of the methods in [“Preparing the Database Layout” on page 101](#).

To continue, click Next.

- 6 In the Configuration Parameter Review screen, review the configuration values you entered on the previous Configuration Utility screens:
  - To go back and make changes, click Previous to back out until you reach the screen with the parameter you need to change.
  - To accept the values you input with no changes, click Finish.

The following message appears:

To apply the configuration now, press “OK”.

To apply the configuration later, press “Cancel”.

The command line to apply the configuration later is

```
C:\sea7xx\siebsrvr\bin\siebug.exe /m master_install_mf.ucf
```

**NOTE:** The last line of the prompt is generated dynamically and might vary based on the location of your Siebel Server installation.

If a program or system error occurs and you need to rerun the Siebel Upgrade Wizard, you can do so, starting at the point at which the wizard failed, by navigating to \siebsrvr\bin and entering the following command:

```
siebug.exe /m master_install_mf.ucf
```

- 7 When the Database Server configuration wizard displays the items to be executed or imported, click OK to start generating DDL.

- This step completes the installation option Run DDL Automatically.
- If in [Step 3 on page 115](#) you chose the installation option Generate DDL Into Files, the utility displays the following message after it generates the schema.sql file that contains the DDL to create the Siebel Schema:

Pause #1: Please create Siebel Schema using schema.sql and ddlview.sql located in the Output Directory. Once the objects are created, please select Yes. To stop now and continue installation later please select No. (To resume at a later time, please start the Upgrade Wizard from the command line using option /m master\_install\_mf.ucf).

The DDL is not executed at this time; it is simply generated to a file. (A message indicates that DDL is being executed; this message refers to execution of the generation of DDL statements and the process of writing the statements to a file.)

- When you are prompted to create the Siebel Schema and views, click Yes to continue.

- 8 Transfer schema.sql and ddlview.sql to OS/390 and have your DBA apply them to create the Siebel Schema.

For DBA information on how to apply the Siebel Schema on the DB2 host, see [“Applying DDL on the DB2 Host” on page 95](#).

- 9 Indicate which action you want to take:

Pause #2: Please ensure that the schemvld.sql is empty (which means that Siebel Schema was created successfully). If so, please continue with the UDFs deployment: ftp the files to install UDF functions. (To resume, please start the Upgrade Wizard from the command line using the option /m master\_install\_mf.ucf).

- Only if your DBA has already created the Siebel Schema, click Yes to populate it. (The execution of the DML does not require DBA authority.)

**CAUTION:** When you click Yes, it means you proceed to populate the schema with seed data, and do other manipulations on Siebel tables. The operation will fail if the Siebel Schema has not yet been created, because there is no schema to populate.

- If you want to quit the DB Server Configuration Utility until your DBA has had an opportunity to create the Siebel Schema, click No. You can relaunch the wizard at any time to continue with this step.

When you have successfully completed the step of populating the seed data, you are ready to import the Siebel Repository.

**TIP:** Locate the file master\_install\_mf.ucf in the binary subdirectory of your Siebel Server installation directory and print it out, so that you have a record of the values you entered.

## Reviewing the Log Files for Windows

The database server component installation creates a number of log files, such as UpgWiz.log, UpgWiz\_01.log (the numeric suffix incrementing for subsequent log files), within the log subdirectory of the siebsrvr installation directory. Review the log files for any errors.

When you install your base language pack, it creates a special log file named dataimp\_prim\_lang.log.

The main log file, produced by the Siebel Upgrade Wizard (UpgWiz.log), does not contain errors returned by applications. Additional information about the configuration process can be found in the sw\_cfg\_utl.log file. Depending on the operation you performed, the ddl\_ctl.log file reflects any errors.

To find additional operation-specific log files, open UpgWiz.log using any text editor and search for a “.log” string. This shows the names of log files produced during the operation; for example, if the configuration of the Database Server fails, the UpgWiz.log file contains the following information:

```
Generi cLog Generi cError 1 2002-02-28 15:51:24 (err=1) was returned by application
(D:\sea77\siebsrvr\bin\ddl imp.exe ...)
```

```
Generi cLog Generi cError 1 2002-02-28 15:51:24 Execute file action failed
(err=Launching the application)
```

In an example such as the one above, the ddlimp.log file contains a detailed error message:

DDLIMP-ERR-1071: Unable to build ddl statement "add col" (Open unloadtbl.jcl)

A Siebel utility called `htmlgenm` can also be helpful by creating an HTML file with the summary report of all steps performed by the Upgrade Wizard together with the information on log files.

Three options exist for the use of `htmlgenm`:

|                         |                      |
|-------------------------|----------------------|
| /M Master UCF file      | Default: master.ucf  |
| /H Summary HTML file    | Default: summary.htm |
| /L Siebupg.exe LOG file | Default: UpgWiz.log  |

For example, if you want to see the HTML results from the database installation, the following command generates the `summary.htm` file in the `log` subdirectory:

```
htmlgenm /m master_install_mf.ucf
```

## Performing a Custom Installation Under UNIX

Perform the following procedure if you are installing Siebel eBusiness Applications on the UNIX operating system.

### *To perform a custom installation under UNIX*

- 1 Follow the steps previously documented for performing a standard installation on the UNIX operating system. Continue until you reach [Step 15 on page 92](#), and enter 2 for Customized Install.
- 2 Enter the number that corresponds to the mechanism you want to use for installation:
  - 1 - Generate DDL Into Files. Generates DDL required to create the Siebel Schema. Proceed to [Step 3](#).

**NOTE:** The DDL is not executed at this time; it is simply generated to a file. A message indicates that DDL is being executed; this message refers to execution of the generation of DDL statements and the process of writing the statements to a file.

You must later use Siebel-provided scripts, your own FTP, or a similar file transfer program to transport the DDL (`schema.sql` and `ddlview.sql`) to the DB2 host, where the DBA executes it, using customary methods, for example, `SPUFI` or `DSNTEP2`.

- 2 - Run DDL automatically. Select this to apply the DDL and DML required to create the Siebel Schema directly against the database. The Siebel application uses ODBC to apply the required DDL to the DB2 host. Proceed to [Step 4](#).
- 3 Enter the path to the output directory for the DDL files you will transfer to your zSeries machine, or accept the default directory:

```
dbsrvr/db2390/dboutput/install
```

**NOTE:** If you plan to use a different output directory to hold the DDL than the Siebel default, you must create it before selecting this option.

- 4 Enter the name of the customized storage control file that you created and saved earlier; this file will be used to configure storage on DB2, including the full path to its location.

This can be either a Siebel-provided template or a storage control file that you have already customized using any of the methods in [“Preparing the Database Layout” on page 101](#).

- 5 Review the list of the parameters you input. If they are correct, enter Y to generate the Siebel Schema.

If Run DDL automatically is chosen, the Siebel Schema is applied directly to DB2. If Generate DDL into Files is chosen, the schema.sql and ddlview.sql files are created.

## Reviewing the Log Files for UNIX

The database server component installation creates a number of log files, such as srvrupgwiz.log, srvrupgwiz1.log (which increments to additional logs) within the log subdirectory of the siebsrvr installation directory, which you must review for any errors.

When you install your base language pack, it creates a special log file with the name dataimp\_prim\_lang.log.

The main log file, produced by the Siebel Upgrade Wizard (Srvrupgwiz.log), does not contain errors returned by applications. Additional information about the configuration process can be found in the sw\_cfg\_utl.log file. Depending on the operation performed, ddl\_ctl.log file reflects any errors:

To find additional operation-specific log files, open Srvrupgwiz.log, using any text editor, and search for a “.log” string.

This shows the names of log files, produced during the operation. For example, if the configuration of the Database Server fails, the Srvrupgwiz.log contains the following information:

```
GenericLog GenericError 1 2002-02-28 15:51:24 (err=1) was returned by application
(D:/sea77/siebsrvr/bin/ddl imp.exe ...)
```

```
GenericLog GenericError 1 2002-02-28 15:51:24 Execute file action failed
(err=Launching the application)
```

In the previous example, the ddlimp.log file contains a detailed error message:

```
DDLIMP-ERR-1071: Unable to build ddl statement "add col " (Open unloadtbl.jcl)
```

The log files might include errors that are expected and benign. Compare any error messages found in the log files to the sample error messages in the errors.rtf file that is located in the db2390 subdirectory of your dbsrvr installation directory. (If a log file is not listed in the errors.rtf file, then there are no acceptable error messages for that log file.) No further action is required if the log files contain only errors listed in the errors.rtf file.

**NOTE:** Only one of each type of error occurring in a particular log file appears in the errors.rtf file.

If you find errors not listed in the errors.rtf file, correct the condition that caused the errors, and rerun the Upgrade Wizard. The wizard restarts from the point where it left off.

Do not review error numbers alone, because error numbers might change after installation of a new driver version. Instead, compare the actual error descriptions to find out which are acceptable errors for this platform.

**CAUTION:** Although other errors are rarely encountered, this review is critical. Certain errors, such as a failure to create indexes, might result in performance problems or anomalous behavior in Siebel eBusiness Applications.

## Rerunning the Installation

If you need to rerun the installation from the beginning, either after completing an installation run or after starting but not finishing the run, delete the `driver_install_mf` subdirectory from the `upgrade` subdirectory of your `siebsrvr` installation directory.

You might need to rerun the installation when you generate new DDL for the Siebel Schema based on a new or modified storage control file.



# Siebel Repository Import, Export, and Postinstallation Tasks

This chapter describes how to import and export the Siebel Repository, and provides both mandatory and optional post-installation procedures. For an overview of the involved tasks, see [Table 15](#).

The chapter consists of the following topics:

- [“About the Database Server Installation and Configuration”](#)
- [“Importing and Exporting the Siebel Repository” on page 122](#)
- [“Reviewing the Log Files for Repository Import” on page 128](#)
- [“Postinstallation Tasks for the Siebel Repository” on page 129](#)

## About the Database Server Installation and Configuration

Table 15. Database Server Installation and Configuration Tasks

| Who Performs It?             | Task                                                                                                                                                                                                                       |
|------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Siebel administrator         | Import the Siebel Repository. See <a href="#">“Importing and Exporting the Siebel Repository” on page 122</a> .                                                                                                            |
|                              | Review the repository import log. See <a href="#">“Reviewing the Log Files for Repository Import” on page 128</a> .                                                                                                        |
|                              | Perform any necessary troubleshooting. See <a href="#">“Troubleshooting the Repository Import” on page 129</a> .                                                                                                           |
| Database administrator (DBA) | Grant table privileges and load statistics. See <a href="#">“Granting Table Privileges” on page 129</a> .                                                                                                                  |
| Siebel administrator and DBA | Run dbchk to validate the physical schema against the Siebel Repository. See <a href="#">“Validating the Siebel Schema” on page 130</a> .                                                                                  |
|                              | If you are deploying multiple languages, install multilingual seed data. See <a href="#">“Installing Multilingual Seed Data” on page 132</a> .                                                                             |
|                              | If you are deploying multiple languages, import multilingual seed data to your repository table rows. See <a href="#">Importing a New Language to Your Repository in “Installing Multilingual Seed Data” on page 132</a> . |
| DBA                          | Populate the Siebel File System. See <a href="#">“Populating the Siebel File System” on page 137</a> .                                                                                                                     |

# Importing and Exporting the Siebel Repository

Repository import is the final step in the Siebel Schema installation process.

When you import the Siebel Repository, you populate all the repository tables in the Siebel Database Server with Siebel eBusiness Applications, Release 7.x application objects.

Regardless of how many Siebel eBusiness Applications you are using (for example, Siebel Sales, Siebel Service, Siebel Marketing), you load data into the repository tables only once per installation.

The Siebel Repository can also be exported into a platform-independent file that can be sent to Siebel Technical Support for analysis in case of problems that cannot be diagnosed by telephone or email.

## *To import the Siebel Repository under Windows*

- 1 Launch the Database Server configuration wizard.
- 2 In the Gateway Name Server Address screen, type the following values as you recorded them in your copy of [Appendix A, "Deployment Planning Worksheet."](#)

- **Gateway Name Server Address.** The alias of the host on which you installed the Siebel Gateway Name Server.
- **Enterprise Server Name.** The name you gave to your Enterprise Server, for example, siebel.

To continue, click Next.

**CAUTION:** If you subsequently add a new language (after you complete your initial installation with your initial set of languages), leave the Gateway Name Server Address and Enterprise Server Name fields blank.

- 3 On the Siebel Server Directory screen, perform either of the following tasks:
  - Accept the default value displayed in the Siebel Server Directory field. (This is the *SIEBSRV\_ROOT* directory, for example, C:\sea7xx\siebsrvr.)
  - Click Browse to select an alternate directory path.

To continue, click Next.

- 4 On the Siebel Database Server Directory screen, perform either of the following tasks:
  - Accept the default path displayed in the Siebel Database Server Directory field. (This is the *DBSRVR\_ROOT* directory, for example, C:\sea7xx\dbsrvr.)
  - Click Browse to select an alternate directory path.

To continue, click Next.

- 5 On the Database Platform screen, select the database platform you use, in this case, IBM DB2 UDB for z/OS and OS/390.

To continue, click Next.

- 6 On the Siebel Database Operation screen, select Import/Export Repository, and click Next.

- 7 On the Select Repository Operation screen, you can choose from the following options. Select the Import Repository option.

- **Import Repository.** To import the Siebel Repository for the first time with a base language, select this option. Proceed to [Step 8](#).
  - **Add Language to an Existing Repository.** If you already imported your Siebel Repository and its base language, select this option to add another language to the repository. See [“Importing a New Language to Your Repository” on page 134](#).
- NOTE:** You must have already imported your base-language repository before you can add a new language.
- **Export Repository.** Exports the Siebel Repository into a platform-independent file that can be sent to Siebel Technical Support for analysis if needed.

**NOTE:** Repository export can also be used to replicate repositories.

To continue, click Next.

- 8 On the Import Selection screen, indicate the following:

- To import a standard Siebel 7 repository, click the uppermost radio button.
- To import a Siebel 7 repository that you have customized to a new environment, click the bottom radio button.

**NOTE:** Select Import Custom Repository when you are importing a multilingual repository from a test or development environment. This imports *all* languages to your target repository.

To continue, click Next.

- 9 On the ODBC Data Source Name screen, enter the name of the ODBC data source, for example, SiebSrvr\_SIEBEL, or enter the database alias you prefer to use for the data source. This alias should have been previously registered within DB2 Connect.

**NOTE:** If you want to work with information on a different database server than the one you currently point to, you can enter the ODBC name of that database server, as long as you have previously registered this as an ODBC data source within DB2 Connect.

The Siebel Server installation process automatically creates the data source, using the syntax: SiebSrvr\_EnterpriseServerName.

To continue, click Next.

- 10 In the Database User Name screen, indicate the following:

- **Database User Name.** Type the ID used to log into the Siebel Database Server. This user ID should be part of the security group with authorization to set the current SQLID to the schema name.
- **Database Password.** Type the password for the ID used to log into the Siebel Database Server.
- **Database Password (confirm).** Retype the password to confirm it.

To continue, click Next.

**11** On the Siebel Schema Qualifier screen, indicate the following values for your Siebel Schema:

- **Siebel Schema Qualifier.** Type the up to eight-character ID that identifies the Siebel Schema owner, for example, SIEBTO.

**NOTE:** This ID must be in uppercase, must start with a letter, and cannot include any special characters. This is the SQL Authorization ID stored in the CURRENT SQLID special register.

- If you are importing your repository for the first time, the Import Repository Name screen appears. Proceed to [Step 12](#).
- If you are *exporting* your repository, the Export Repository Name screen appears. Proceed to [Step 13](#).

To continue, click Next.

**12** On the Import Repository Name screen, indicate the following:

- **Import Repository Name.** Enter the designated name for this Siebel Repository.
- **Repository File Name.** If you are importing your repository for the first time, this is the second field appearing on the screen. Accept the default installation path and file name for this repository or type another valid installation path.

To continue, click Next. Proceed to [Step 14](#).

**13** On the Export Repository Name screen, type the following values:

- **Export Repository Name.** Accept the default name (Siebel Repository) or type another valid name.
- **Repository File Name.** Accept the default installation path and file name for this repository or type another valid installation path.

To continue, click Next.

**14** On the Log Output Directory screen, accept the default name (imprep) or type another valid name.

**15** On the Configuration Parameter Review screen, review the configuration values that you entered on the previous configuration screens:

- To go back and make changes, click Previous to back out until you reach the screen with the parameter you need to change.
- To accept the values you input with no changes, click Finish.

The runnow message box appears with the prompt:

To apply the configuration now, press "OK".

To apply the configuration later, press "Cancel".

The command line to apply the configuration later is  
C:\sea7x\si ebsrvr\bin\siebug.exe /m master\_impreg.ucf

**NOTE:** The last line in the prompt is generated dynamically based on the location of your Siebel Server installation.

If a program or system error occurs and you need to rerun the Siebel Upgrade Wizard, you can do so, starting at the point at which the wizard failed, by navigating to \siebsrvr\bin and entering:

siebug.exe /m master\_impreg.ucf

- 16** When the Database Server configuration wizard displays the items to be executed or imported, click OK to begin.

A window appears, displaying information about the repository import activities. Afterwards, a message appears when the repository import is complete.

To verify that the import was successful, review the log files. See [“Reviewing the Log Files for Repository Import” on page 128](#).

**TIP:** Locate the file master\_impreg.ucf in the binary subdirectory of your Siebel Server installation directory and print it out, so that you have a record of the values you entered.

### ***To import the Siebel Repository under UNIX***

- 1** Navigate to `$SIEBEL_HOME` and source environment variables, using one of the following commands, depending on the type of shell you use:

#### **Korn or Bourne shell**

```
. ./siebenv.sh
```

**TIP:** Make sure there is a space between the initial period and `./siebenv.sh`.

#### **C shell**

```
source siebenv.csh
```

where:

`$SIEBEL_HOME` is the location of the Siebel Server root directory.

- 2** Verify whether the values `$SIEBEL_ROOT` and `LANGUAGE` are correctly set or are empty by entering:

```
env
```

If the variables are incorrect or missing, reset them using one of the following commands, as appropriate to the shell you use:

#### **Korn or Bourne shell**

```
SIEBEL_ROOT=New Value ; export SIEBEL_ROOT
```

```
LANGUAGE=New Value; export LANGUAGE
```

**C shell**

```
setenv SIEBEL_ROOT New Value
```

```
setenv LANGUAGE New Value
```

- 3** Go to `$SIEBEL_ROOT/bin` and enter:

```
dbsrvr_config.ksh
```

This launches the Database Server configuration wizard.

- 4** Review the values of the following environment variables and confirm whether the settings are correct by entering either Y or N.

`SIEBEL_ROOT` = path of your Siebel installation directory

`LANGUAGE` = language in which you are installing

**NOTE:** If either the `SIEBEL_ROOT` or `LANGUAGE` value is not set or is incorrect, you must correct them before proceeding.

- 5** Verify and accept the default for the path of your Siebel Server root directory and press ENTER.

- 6** Verify and accept the default for the path of your database server root directory and press ENTER.

- 7** Select your RDBMS platform, in this case, IBM DB2 for z/OS and OS/390.

- 8** From the Siebel Database Operation menu, choose Import/Export Repository.

- 9** Choose one of the following options that corresponds to the procedure you want to complete:

1 - Import Repository. To import the Siebel Repository for the first time with a base language, select this option. Proceed to [Step 10 on page 127](#).

2 - Add Language to an Existing Repository. If you already imported your Siebel Repository and its base language, select this option to add another language to the repository. See ["Importing a New Language to Your Repository" on page 134](#).

**NOTE:** You must have already imported your base-language repository before you can add a new language.

3 - Export Repository. Exports the Siebel Repository into a platform-independent file that can be sent to Siebel Technical Support for analysis if needed. Proceed to [Step 11 on page 127](#).

**NOTE:** Repository export can also be used to replicate repositories.

If you selected Import Repository, the Import Selection screen appears.

**10** Specify the applicable import option:

1 - Import Siebel Repository. To import one additional language to your repository, enter 1. See [“Importing a New Language to Your Repository” on page 134](#).

2 - Import Custom Repository. To import *all* languages to your repository, enter 2. Proceed to [Step 11](#).

**NOTE:** Option 2 is used when you are importing a multilingual repository from a test or development environment.

**11** Enter the number corresponding with the language in which you want to run your database. (This is known as your base language.)

**12** Enter the name of the ODBC data source, for example, SiebSrvr\_siebel, or enter the database alias you prefer to use for the data source. This alias should have been previously registered within DB2 Connect.

**NOTE:** If you want to work with information on a different database server than the one you currently point to, you can enter the ODBC name of that database server, as long as you have previously registered this as an ODBC data source within DB2 Connect.

The Siebel Server installation process automatically creates the data source, using the syntax: SiebSrvr\_EnterpriseServerName.

**13** Specify the Database User Name.

This is the ID allowed to log into the Siebel Database. This user ID should have authorization to set the CURRENT SQLID, for example, SADMIN.

**14** Enter the database password.

**15** Indicate the following values:

- Specify the Siebel Schema Qualifier, for example, SIEBTO.

This is the up to eight-character (all uppercase) identifier that designates the Siebel Schema. This is also an authorization ID. The Schema Qualifier must be in uppercase, must start with a letter, and cannot contain special characters.

- Specify the Security Group ID / Grantee or accept the default (SSEROLE) by pressing ENTER.

This is the group to whom schema access is being granted.

**16** Enter the name of your import or export repository, or accept the default, Siebel Repository.

**17** Enter the name of your repository file or accept the default, mstrep.dat.

The utility displays the values entered so far and prompts you as to whether you want to run the Siebel Upgrade Wizard.

- If the values are correct, launch the Siebel Upgrade Wizard (srvrupgwi z) by entering Y.

- To change the values you previously entered, rerun the Database Server configuration wizard by entering `dbsrvr_config.ksh`. This allows you to reconfigure with different values.

**NOTE:** If a program or system error occurs and you need to rerun the Siebel Upgrade Wizard, you can do so, starting at the point at which the wizard failed, by navigating to `$SIEBEL_ROOT/bin` and entering: `srvrupgwiz /m master_impreg.ucf`

If the repository import was successful, it displays a message that the repository has been imported. Otherwise, see the log files.

## Reviewing the Log Files for Repository Import

The repository import process creates a number of log files within the subdirectory `siebsrvr\log` (`$SIEBEL_ROOT/log` under UNIX) that you must review for any errors. The log file names for subsequent processes include a numeric suffix that increments for additional logs files; for example, `UpgWiz.log`, `UpgWiz_01.log`, `UpgWiz_02.log` (under Windows) or `srvrupgwiz.log`, `srvrupgwiz1.log`, `srvrupgwiz2.log` (under UNIX). Further log files are created when the Upgrade Wizard encounters a problem and the user attempts a retry.

When you import a repository with a new language, it creates the following special log files:

- `impreg_prim.log`
- `impreg_lang.log`

For more details on logs, refer to the discussion under [“Reviewing the Log Files for Windows” on page 117](#) and [“Reviewing the Log Files for UNIX” on page 119](#).

## About Acceptable Errors on the errors.rtf File

The log files might include errors that are expected and benign. Compare any error messages found in the log files to the sample error messages in the `errors.rtf` file that is located in the `db2390` subdirectory of the `dbsrvr` installation directory. (If a log file is not listed in the `errors.rtf` file, then there are no acceptable error messages for that log file.) No further action is required if the log files contain errors listed in the `errors.rtf` file.

**NOTE:** Only one of each type of error occurring in a particular log file appears in the `errors.rtf` file.

If you find errors not listed in the `errors.rtf` file, correct the condition that caused the errors, and rerun the Upgrade Wizard. If you cannot determine the cause of the errors on your own, contact Siebel Technical Support.

Do not review error numbers alone, because error numbers might change after installation of a new driver version. Instead, compare the actual error descriptions to find out which are acceptable errors for this platform.

**CAUTION:** Although other errors are rarely encountered, this review is critical. Certain errors, such as a failure to create indexes, might result in performance problems or anomalous behavior in Siebel eBusiness Applications.



## Troubleshooting the Repository Import

One problem that can occur at this stage is the failure of the Siebel Repository import midway through the process. If this occurs, you must clean up the data from the failed import using Siebel Tools to delete the repository.

# Postinstallation Tasks for the Siebel Repository

Postinstallation tasks consist of the following activities:

- “Granting Table Privileges”
- “Loading Statistics”
- “Validating the Siebel Schema” on page 130
- “Installing Multilingual Seed Data” on page 132
- “Importing a New Language to Your Repository” on page 134
- “Populating the Siebel File System” on page 137

## Granting Table Privileges

The database administrator must edit and execute the file `grantstat.sql`, located within the `db2390` subdirectory of your Siebel Database Server installation directory (`dbsrvr`), to grant the Siebel group ID additional, required privileges on EIM tables.

### *To edit and run the `grantstat.sql` script*

- 1 Locate the `grantstat.sql` script and edit it by replacing &1 with the Siebel Schema Qualifier ID that you recorded in [Appendix A, “Deployment Planning Worksheet.”](#)
- 2 Substitute &2 with the security group ID or with any user ID.
- 3 Execute the `grantstat.sql` script using any method you are used to.

**NOTE:** If you are using multiple security groups, you must grant this privilege to each group.

## Loading Statistics

Execution of the script `loadstat.sql` loads statistics required for operation of the Siebel application into the DB2 catalog.

### *To edit and run the `loadstat.sql` script*

- 1 Locate the script `loadstat.sql` within the `db2390` subdirectory.

- 2 Follow the instructions documented inside loadstat.sql to edit the file, making sure to change the variable *SIEBELTABLECREATOR* to the CREATOR for the Siebel Schema as defined in the DB2 catalog.
- 3 Execute the script using any method you are used to.

## Validating the Siebel Schema

After installation and repository import, you should compare the physical database schema with the repository to make sure that there are no inconsistencies. This is accomplished using the Siebel Server utility dbchck.exe (Windows) or dbchck (UNIX), located within the binary subdirectory of your Siebel Server installation directory.

You can use the dbchck utility to validate data relationships, including foreign keys and the list of values. You can also use this utility when you have made changes to the extensibility of your Siebel database.

The most popular mode in which to run dbchck is using the option */di ct /al l*. This option provides a comprehensive log file that lists all of the discrepancies between the logical data model defined in the repository you specify and the physical database schema in the Siebel database you run the utility against.

### To run dbchck

- 1 From the binary subdirectory of your Siebel Server installation directory, locate dbchck.exe or dbchck, depending on your OS platform.
- 2 If your midtier platform is Windows, delete the diccache.dat file before running dbchck. But be sure that no EIM or Siebel Remote operations are running before deleting it.

The dbchck utility creates a new diccache.dat file as a first step before proceeding with the integrity check. By deleting the dictionary cache file before starting, you make sure that dbchck validates against the Siebel Repository you specify.

**NOTE:** If you had to stop any EIM or Siebel Remote processes, you can safely restart them after dbchck has run.

- 3 Source environment variables as appropriate for your platform, as described earlier in this chapter, using either siebenv.bat (for Windows) or siebenv.sh or siebenv.csh (for UNIX).

Environment variable scripts are located in the binary subdirectory of your Siebel Server installation directory.

- 4 Source the database profile.
- 5 Run dbchck using the following syntax:

```
dbchck /u SADMIN /p password /t SIEBTO /r "Your Siebel Repository Name" /l dbchck.log
/di ct /al l /s ODBC data source
```

where:

- *password* = login password to the database.

- *Your Siebel Repository* = the repository that you want to compare against the physical data model; in the installation, the default value is "Siebel Repository".

**NOTE:** You must specify your repository name within quotation marks (") after the repository name parameter (/r).

- *ODBC data source* = the ODBC data source applicable to the repository.
- (/t) = the Siebel Schema Qualifier.

**NOTE:** To see a list of all dbchk parameters with their descriptions, use option /h.

Any discrepancies found appear on the screen. Detailed information is written to a log file in the binary directory of your Siebel Server installation directory (siebsrvr).

**NOTE:** The log file name is whatever name you specified after the log file parameter (/l). In the example above, the file is named dbchk.log.

- 6 Review the log file generated as a result of running this script. Any discrepancies are flagged as failures.
- 7 You should investigate all discrepancies.

## Investigating Discrepancies in the dbchk Log

After completion of the dbchk.exe script, the log file generated may contain discrepancies noted as failures. However, some failures are expected due to the index length restriction.

One expected error follows:

- Failure: Dictionary for EIM\_ACCOUNT2 different from physical schema  
Dictionary different from physical schema for index EIM\_ACCOUNT2\_T16:  
  
dictionary: (IF\_ROW\_BATCH\_NUM ACC\_NAME ACC\_BI OP\_PROD\_BI OP\_PROD\_NAME  
OP\_PROD\_VEN\_NAME ACC\_LOC OP\_PROD\_VEN\_LOC OP\_INSTALL\_DT)  
physical: (IF\_ROW\_BATCH\_NUM ACC\_NAME ACC\_BI OP\_PROD\_BI OP\_PROD\_NAME)

These messages in the dbchk.log file are expected and are caused by the index length restriction of 255 bytes that applies to indexes created on the OS/390 platform. The truncated indexes will not necessarily cause any issues as some of these tables with truncated indexes will not be used in the application. However, the truncated indexes should be reviewed to see whether any of these need to be addressed based on your business requirement.

Some errors noted in the dbchk.log file are not acceptable and should be addressed. An example of such an error follows:

- Dictionary column FIN\_PERIOD\_ID not in physical schema!  
Dictionary different from physical schema for column  
dictionary: REVISSED\_COST numeric(22, 7) null  
physical: REVISSED\_COST decimal(10, 0) null

This error is due to a mismatch in the data type of the column in the logical and physical schema. Columns in logical and physical schema should always be synchronized as far as the column definition is concerned.

You will need to correct this mismatch in Siebel Tools in order to synchronize with the physical schema. Once this is done, run dbchck.exe again.

For more information about using and troubleshooting dbchck.exe, see the FAQ 1128 document and the Troubleshooting Steps 20 document on Siebel SupportWeb.

## Installing Multilingual Seed Data

If your organization deploys internationally and, therefore, requires data to be in multiple languages, you must install multilingual seed data (for example, lists of views, responsibilities, or system preferences). To do this, you add new language packs to your database after you have installed the base language for your database server. This populates the List of Values (LOV) with seed data in the new language.

Only after you successfully install seed data in your base language, can you add seed data in other languages to your database. Adding this seed data also adds new LOV data in the new language.

**CAUTION:** You cannot add secondary languages to the Siebel database for an Enterprise Server unless you have already installed them on the associated Siebel Server.

### *To install multilingual seed data under Windows*

- 1 Follow the instructions under [“Siebel Database Server Software Installation” on page 73](#), using the DVD for the new language that you want to install.
- 2 In the Existing Installations Found screen, select the installation to which you want to add the language.
- 3 When installation is complete, launch the Database Server configuration wizard.
- 4 When the Select Installation Operation screen appears ([Step 7 on page 86](#)), select Add a language to an existing Siebel Database, and then click Next.

**NOTE:** To add seed data in a new language to your database, you must have already imported your repository in its base language.

The Base Language screen appears and displays the languages you have installed for your Siebel Database.

- 5 In the Base Language screen, specify which of the languages you installed is the one in which you want to primarily run your database. (This is known as your base language.)

To continue, click Next.

**NOTE:** The Language Selection screen appears only if you have installed the files for multiple Siebel Language Packs. Installation of multiple Language Packs can occur either during your initial installation of Siebel 7 or at a later time.

- 6 In the Language Selection screen, specify the new language you are adding for this database.

To continue, click Next.

The ODBC Data Source Name screen appears.

- 7 Respond to the remaining prompts as you did when you originally completed the procedure, [“Installing the Siebel Schema” on page 84](#), until the Repository Name screen appears.
- 8 Type the name of your Siebel Repository or accept the displayed default name.  
To continue, click Next.
- 9 In the Configuration Parameter Review screen, review the configuration values you entered on the previous configuration screens:
  - To go back and make changes, click Previous to back out until you reach the screen with the parameter you need to change.
  - To accept the values you input with no changes, click Finish.

A message box appears with the prompt:

To apply the configuration now, press “OK”.

To apply the configuration later, press “Cancel”.

The command line to apply the configuration later is

```
C:\sea7xx\si ebsrvr\bin\siebug.exe /m master_install_lang.ucf
```

**NOTE:** The last line in the prompt is generated dynamically based on the location of your Siebel Server installation.

If a program or system error occurs and you need to rerun the Siebel Upgrade Wizard, you can do so, starting at the point at which the wizard failed, by navigating to \siebsrvr\bin and entering:

```
siebug.exe /m master_install_lang.ucf
```

The Database Server configuration wizard displays the items to be executed or imported.

- 10 To begin, click OK.  
A window appears, displaying information about installation activities. Afterwards, a message appears when the installation is complete.  
To verify that the installation was successful, review the log files. See [“Reviewing the Log Files for Repository Import” on page 128](#).  
**TIP:** Locate the file `master_install_lang.ucf` in the binary subdirectory of your Siebel Server installation directory and print it out, so that you have a record of the values you entered.
- 11 When you have finished configuration using the Database Server configuration wizard, you must enable the multilingual list of values (MLOV) capability within Siebel eBusiness Applications. For information on how to do this, see *Global Deployment Guide*.
- 12 After you enable MLOV capability as described in *Global Deployment Guide*, proceed to [“Importing a New Language to Your Repository” on page 134](#).

### ***To install multilingual seed data under UNIX***

- 1 Follow the UNIX instructions under [“Siebel Database Server Software Installation” on page 73](#), using the DVD for the new language that you want to install.

- 2 Follow the instructions for UNIX under [“Performing a Standard Installation” on page 84](#) and when prompted for the installation operation ([Step 9 on page 92](#)), enter 2 to Add New Language to an Existing Database.
- 3 At the Language Selection prompt, specify which of the displayed languages you are adding on top of your base language for this database.
- 4 At the Base Language prompt, specify which language you want as your base language.
- 5 Respond to the remaining prompts as you did when you originally completed the procedure for [“Installing the Siebel Schema” on page 84](#) whether you chose standard or custom installation.
- 6 When you are prompted for the repository name, accept the default (Siebel Repository) or enter another name.
- 7 Review the values for accuracy.

- If the values are correct, launch the Siebel Upgrade Wizard (srvrupgwiz) by entering Y.

**NOTE:** If a program or system error occurs and you need to rerun the Siebel Upgrade Wizard, you can do so, starting at the point at which the wizard failed, by navigating to `$SIEBEL_ROOT/bin` and entering:

```
srvrupgwiz /m master_install_lang.ucf
```

- To change the values you previously entered, rerun the Configuration Wizard by entering `dbsrvr_config.ksh`. This allows you to reconfigure with different values.

The Database Server configuration wizard displays the items to be executed or imported.

- 8 To begin, click OK.

A message appears when the installation is complete.

To verify that the installation was successful, review the log files. See [“Reviewing the Log Files for Repository Import” on page 128](#).

**TIP:** Locate the file `master_install_lang.ucf` in the binary subdirectory of your Siebel Server installation directory and print it out, so that you have a record of the values you entered.

- 9 When you have finished configuration using the Configuration Wizard, you must enable the multilingual list of values (MLOV) capability within Siebel eBusiness Applications. For information on how to do this, see *Global Deployment Guide*.
- 10 After you enable MLOV capability as described in *Global Deployment Guide*, proceed to [“Importing a New Language to Your Repository.”](#)

## Importing a New Language to Your Repository

After you successfully import your Siebel Repository in its base language, you can add additional languages to it.

By adding a new language to your repository, you populate rows of localizable information, which allows Siebel eBusiness Applications to better operate in the new language.

Regardless of how many Siebel eBusiness Applications you are using (such as Siebel Sales, Siebel Service, Siebel Marketing), you perform this step only once for each language you want to install.

**CAUTION:** If you subsequently add a new language (after you complete your initial installation with your initial set of languages), in the Database Server configuration wizard, leave the Gateway Name Server Address and Enterprise Server Name fields blank.

### *To import a repository in a secondary language under Windows*

- 1 After installing multilingual seed data ([“Installing Multilingual Seed Data” on page 132](#)) launch the Database Server configuration wizard.
- 2 Follow the steps in [“Importing and Exporting the Siebel Repository” on page 122](#) to [Step 7 on page 123](#) (Select Repository Operation) and select:  
Add Language to an Existing Repository  
To continue, click Next.
- 3 In the Language Selection screen, specify the new repository language you are adding.  
To continue, click Next.
- 4 Follow the remaining procedures in [“Importing and Exporting the Siebel Repository” on page 122](#) up to [Step 12 on page 124](#).
- 5 In the Repository Name screen, indicate the following:
  - **Import Repository Name.** Type the designated name for this Siebel Repository or accept the default.
  - **Localized Repository Name.** If you are importing a secondary language repository on top of your base-language repository, this field appears in place of the Repository File Name field. Accept the default installation path and file name for this repository or type another valid installation path.To continue, click Next.
- 6 In the Configuration Parameter Review screen, review the configuration values you entered on the previous configuration screens:
  - To go back and make changes, click Previous to back out until you reach the screen with the parameter you need to change.
  - To accept the values you input with no changes, click Finish.A message box appears with the prompt:  
To apply the configuration now, press “OK”.  
To apply the configuration later, press “Cancel”.

The command line to apply the configuration later is  
`C:\sea7x\si ebsrvr\bin\siebug. exe /m master_impreg_lang. ucf`

**NOTE:** The last line in the prompt is generated dynamically based on the location of your Siebel Server installation.

If a program or system error occurs and you need to rerun the Siebel Upgrade Wizard, you can do so, starting at the point at which the wizard failed, by navigating to `\siebsrvr\bin` and entering:

`siebug. exe /m master_impreg_lang. ucf.`

The Database Server configuration wizard displays the items to be executed or imported.

- 7 To begin, click OK.

A window appears, displaying information about installation activities. Afterwards, a message appears when the installation is complete.

To verify that the installation was successful, review the log files. See [“Reviewing the Log Files for Repository Import” on page 128](#).

**TIP:** Locate the file `master_impreg_lang.ucf` in the binary subdirectory of your Siebel Server installation directory and print it out, so that you have a record of the values you entered.

### ***To import a repository in a secondary language under UNIX***

- 1 Perform the steps described within the UNIX section of [“Importing and Exporting the Siebel Repository” on page 122](#) up to the prompt for Repository Operations.
- 2 At the Repository Operations prompt, enter 2 for Add language to an existing Repository.
- 3 At the Choose Language prompt, enter the number corresponding to the secondary language you are adding.

**NOTE:** Siebel Systems does not support installation of any language other than U.S. English on top of a base Japanese, Simplified Chinese, Traditional Chinese, or Korean Language Pack.

- 4 Answer the remaining prompts with the values that you previously entered when you completed [“Importing and Exporting the Siebel Repository” on page 122](#) for your base language until prompted for the Import Repository Name.
- 5 At the Import Repository Name prompt, accept the default name of the new repository or specify a new name.
- 6 Enter the name of your repository file name or accept the default, `mstrep.dat`.

The utility displays the values entered so far and prompts you as to whether you want to run the Upgrade Wizard.

- If the values are correct, launch the Wizard (`srvrupgwiz`) by entering Y.



- To change the values you previously entered, rerun the Configuration Wizard by entering `dbsrvr_config.ksh`. This allows you to reconfigure with different values.

**NOTE:** If a program or system error occurs and you need to rerun the Wizard, you can do so, starting at the point at which the wizard failed, by navigating to `$SIEBEL_ROOT/bin` and entering:  
`srvrupgwi z /m master_impreg_lang.ucf`

To verify that the installation was successful, review the log files. See [“Reviewing the Log Files for Repository Import” on page 128](#).

**TIP:** Locate the file `master_impreg_lang.ucf` in the binary subdirectory of your Siebel Server installation directory and print it out, so that you have a record of the values you entered.

## Populating the Siebel File System

Specific files needed to run the Siebel File System, such as correspondence templates and Siebel Marketing files, are provided with the Siebel Database Server software. A subdirectory called `files` is created automatically when you install the Siebel Database Server.

The system administrator must populate the appropriate subdirectory of the Siebel File System with these file attachments after installing the Siebel Database Server and before running the Siebel Web Client.

### *To populate the Siebel File System directory*

- 1 Copy the appropriate files from the `files` subdirectory of the Siebel Database Server software to the `att` subdirectory of the Siebel File System.
- 2 Copy the `KB.kb` file from the `files` subdirectory of the Siebel Database Server software to the `cms` directory within the Siebel File System.
- 3 Verify that the files are where they need to be.



# 10 Customizing a Development Environment

This chapter describes procedures for customizing development environments and then migrating those customizations to user-acceptance or production environments on DB2 UDB for z/OS and OS/390.

This chapter consists of the following topics:

- [“About Customizing Your Development Environment”](#)
- [“Siebel Tools Considerations on DB2 UDB for z/OS and OS/390” on page 140](#)
- [“Converting Nonpartitioned Tables to Partitioned Tables” on page 145](#)
- [“Creating Custom Extensions to the Siebel Schema” on page 146](#)
- [“How to Spool DDL Generated by the Apply Button” on page 150](#)
- [“Applying Schema Extensions to the Target Database” on page 150](#)
- [“Cloning an z/OS and OS/390 Database” on page 158](#)
- [“Migrating Rules and Other Data” on page 161](#)
- [“Applying Schema Changes to Other Local Databases” on page 162](#)
- [“Customizing Applications Using Assignment and Workflow Rules” on page 162](#)

## About Customizing Your Development Environment

Customization of Siebel eBusiness Applications within your development environment includes:

- Converting nonpartitioned tables to partitioned tables
- Changing views, business objects, applets, and tables
- Adding new columns to existing tables and adding new tables
- Modifying workflow policies and workflow processes

These tasks are documented in *Using Siebel Tools* and *Siebel Business Process Designer Administration Guide*. However, several procedures and concerns are specific to DB2 UDB for z/OS and OS/390, so they are therefore included in this chapter.

This chapter assumes that you perform your development on either of the following:

- DB2 UDB on a Windows, AIX, or Solaris machine
- DB2 UDB for z/OS and OS/390 on a partition of your zSeries machine reserved for development

In many cases, changes to user-acceptance and production (server) databases are made by a database administrator working within a change-management system. Therefore, this guide includes procedures for generating Data Definition Language (DDL) files that are later applied to databases.

## Siebel Tools Considerations on DB2 UDB for z/OS and OS/390

There are certain considerations and procedures to follow in Siebel Tools when developing applications that run against databases on DB2 UDB for z/OS and OS/390.

### Setting Siebel Tools Configuration File Parameters

The following parameters in the [ServerDataSrc] section of the Siebel Tools application configuration file (tools.cfg) must be set for DB2 UDB for z/OS and OS/390 as shown in [Table 16](#).

Table 16. Siebel Tools Configuration File Parameters

| Parameter     | Value            |
|---------------|------------------|
| TableOwner    | Schema qualifier |
| MaxCursorSize | -1               |
| PrefetchSize  | -1               |

### Setting Database Options

When developing applications on any platform (for example, DB2 UDB) that will eventually be deployed on DB2 UDB for z/OS and OS/390, you must set the proper database options in Siebel Tools. This enables all Siebel Tools features for DB2 UDB for z/OS and OS/390 and validates Siebel objects for z/OS and OS/390 requirements:

- The names of schema objects are limited to 18 characters.
- CHAR columns with length >1 are allowed.
- Users can eliminate unused indexes.
- Users can reduce the size of VARCHAR fields.

#### ***To set database options for DB2 UDB for z/OS and OS/390***

- 1 In Siebel Tools, choose View > Options.  
The Development Tools Options dialog appears.
- 2 On the Database tab, select the Developing for deployment on DB2 for zSeries check box.

## Specifying Table Grouping File Names

The complete paths for table grouping file names (storage control files) must be specified in the Apply dialog box when applying schema changes to a database.

## Inactivating Unused Indexes

The standard Siebel data model has a large number of indexes that can degrade performance on DB2 UDB for z/OS and OS/390. This is because DB2, unlike Oracle and Microsoft SQL Server, stores NULL as one of the indexed values. Each row with a NULL value for an indexed column gets an entry of its record identifier (RID) in the DB2 index. Because many Siebel tables are related to a large number of other tables, there are many indexes on foreign keys.

Where those related tables are not used, DB2 has the indexes for those foreign keys with one entry: NULL. This key entry has a long RID chain comprised of all the RIDs for every row in the table. There is maintenance overhead each time a row is inserted, and even more overhead when it is deleted. Therefore, users of DB2 UDB for z/OS and OS/390 are allowed to inactivate unused indexes using Siebel Tools and drop them in DB2.

**CAUTION:** Before deleting indexes, back up your Siebel Repository and the original storage control and schema.sql files that you installed. Keep track of the indexes you delete. If you notice performance degradation after deleting indexes, review your work and restore the repository, storage control file, and schema.sql. If you need help, contact Siebel Technical Support.

## Reducing Field Lengths of Unique Indexes to Avoid Truncation

Indexes on DB2 UDB for z/OS and OS/390 are limited to 255 characters. If you customize your development environment without setting the database options for DB2 UDB for z/OS and OS/390, your indexes might be truncated upon migration to the DB2 host and could become nonunique. For more information, see [“Setting Database Options” on page 140](#).

When analyzing your business needs, if the unique indexes seem too long, it is recommended that you reduce the field lengths.

## Reducing VARCHAR Field Lengths

DB2 UDB for z/OS and OS/390 pads VARCHAR fields to their maximum lengths in indexes. For greatest flexibility, Siebel applications make extensive use of VARCHAR columns.

This might cause performance degradation for z/OS and OS/390 customers who have a large number of rows in certain tables, such as S\_CONTACT.

The total length of an index equals the combined lengths of all index columns. To reduce the index length, you must reduce the length of participating columns. To make better use of index space, users of DB2 UDB for z/OS and OS/390 are allowed to reduce VARCHAR lengths—where there is no application impact—using Siebel Tools.

You should analyze your business needs at installation or upgrade to determine whether it is necessary to reduce VARCHAR field lengths. Some examples are the following:

- FST\_NAME and LAST\_NAME, VARCHAR(50) columns in S\_CONTACT. These belong to many indexes.
- ADDR, a VARCHAR(200) column in S\_ADDR\_PER.
- NAME, a VARCHAR(100) column in S\_ORG\_EXT.

Be aware that you also must make the same length reductions to columns denormalized from those you have reduced. One example is S\_PER\_RESP.PER\_FST\_NAME, which is denormalized from S\_CONTACT.FST\_NAME.

A denormalized column duplicates the data in a column in another (base) table for performance reasons. The table and column names of the duplicated column are specified in the Denormalization Path property of the Column object definition of the denormalized column. For more information on columns, see *Using Siebel Tools*.

## **LONG Columns on DB2 UDB for z/OS and OS/390 and Other DBMS Platforms**

Siebel uses LONG columns on databases servers to store arbitrarily long character data up to 16350 characters. Examples include scripts, emails, notes, and descriptions. On all RDBMS platforms, except DB2 UDB for z/OS and OS/390, the size of LONG columns is fixed and is independent of the size of the table space in which the table is defined.

On DB2 UDB for z/OS and OS/390, a LONG column defined in the Siebel schema is mapped to a LONGVARCHAR column, which always has a size of 32700 bytes.

On DB2 UDB for z/OS and OS/390, the size of a LONGVARCHAR column varies depending on two factors:

- The size of the table space in which the table is defined.
- The combined byte size of all non-LONGVARCHAR columns of the table.

The size of the LONGVARCHAR column is roughly calculated by the size of the table space subtracted by all non-LONGVARCHAR columns. In other words, a LONGVARCHAR column on DB2 UDB for z/OS and OS/390 would take up the rest of the space available in the table space. For example, a table containing one LONGVARCHAR column defined in a 16-KB table space with a combined byte size for non-LONGVARCHAR columns of 5 KB would result in the LONGVARCHAR column having a size of about 11 KB. For more information on size calculation of LONGVARCHAR columns, refer to vendor documentation on the IBM Web site.

## How Siebel Systems Uses LONG Columns on DB2 UDB for z/OS and OS/390

Creating all LONGVARCHAR columns on DB2 UDB for z/OS and OS/390 with length 16350 would effectively force all tables to a 32-KB tablespace. Doing so will increase buffer pool storage requirements as it needs twice as much storage as a 16-KB tablespace. So, in order to minimize the number of tables created in 32-KB tablespaces, the Siebel installation process by default creates tables with LONGVARCHAR columns in a 16-KB tablespace. This is done by assuming that LONGVARCHAR columns have a logical length of 4046 bytes, which in effect forces the table to be created in at least a 16-KB tablespace.

**NOTE:** Although logically Siebel Systems sets the LONGVARCHAR column to have a logical length of 4046 bytes, the physical length is not specified in the CREATE TABLE statements for the LONGVARCHAR column. So the resulting length of the LONGVARCHAR field is still calculated by the algorithm defined by DB2 UDB for z/OS and OS/390.

By doing this the resulting size of LONGVARCHAR columns will vary between 4046 bytes to 16320 bytes. With this approach, almost all aspects of the Siebel application function without any problems. However, if your implementation involves a usage scenario in which very large notes, descriptions, emails, or other items up to 16350 in length are stored, then you need to resolve this by considering one of the two following choices:

- Move the table to a 32-KB tablespace.
- Use a CLOB column instead of a LONGVARCHAR column.

Before implementing either of the two solutions above, you should first understand the potential issues that may arise from either of the two solutions.

### About Moving the Table to a 32-KB Tablespace

By moving a table to a 32-KB tablespace, the LONGVARCHAR column would have a length up to 30 KB. As mentioned before, doing this will double the storage requirements for bufferpools. Another side effect is that the result set of a join SELECT statement containing a LONGVARCHAR column could be more than 32 KB. When the result set does exceed 32 KB, then you may run into the 32-KB in-memory sort limit of DB2 UDB for z/OS and OS/390. When DB2 UDB for z/OS and OS/390 performs an in-memory sort, the full length of the LONGVARCHAR column is added to the total row width. As a result, you may run into SQL0670N errors.

### About Using a CLOB Column Instead of a LONGVARCHAR Column

By using a CLOB column instead of a LONGVARCHAR column, you will not run into the 32-KB in-memory sort limit of DB2 UDB for z/OS and OS/390. The reason is DB2 UDB for z/OS and OS/390 does not account for the actual length of the CLOB column in the total row width to be sorted in memory. However, although there are no functional limitations, there is impact on performance and storage (DASD). Fetching data from CLOB columns requires extra network flows between the client and database server. This can impact response time. Also, writing data to CLOB columns is slower than writing to LONGVARCHAR columns. So, if the table in question is frequently used, then moving to CLOB columns is not an option.

Using CLOB columns will most likely increase the amount of storage (DASD) needed. The reason is the space for the total width of the CLOB column will be allocated and used even if only one byte of data is in the CLOB column.

IBM and third-party vendor tools have limited or no support for CLOB columns as well. Compression is also not supported for LOB tablespaces.

Aside from performance and storage tradeoffs, perhaps the most important tradeoff for using CLOBs is recovery considerations. Using CLOBs with LOG YES provides a point of forward recovery; however, there is a cost associated with logging the contents of the CLOB column. This additional overhead should also be considered before deciding on using CLOB columns.

## About Defining Siebel Tables with LONG Columns in 32-KB Tablespaces

The following tables are defined in 32-KB tablespaces in the Siebel application. Siebel eBusiness Applications moved these tables to a 32-KB tablespace, because the LONG columns in these tables need as much space as possible.

- S\_BI TMAP\_DATA
- S\_SCHMST\_DBSCPT
- S\_SERVI CE\_SCRPT
- S\_NOTE
- S\_SRC
- S\_DMND\_CRTN\_PRG

## About Using CLOB Columns in the Siebel Application

The following three tables use CLOB columns in the Siebel application. Siebel eBusiness Applications used CLOB columns for these tables because the tables belong to the Siebel repository and read and write activity to them is small. Also, CLOB columns are used if the table could not be put in a 32-KB table space because of the 32-KB in-memory sort limit of DB2 UDB for z/OS and OS/390.

- S\_DMND\_CRTN\_PRG
- S\_NOTE
- S\_BI TMAP\_DATA

## Changing LONGVARCHAR Datatype to VARCHAR Datatype

To avoid ambiguity in the actual length of the underlying field, a LONGVARCHAR datatype can be changed to a VARCHAR datatype on the physical schema.



### ***To change a LONGVARCHAR Datatype to a VARCHAR Datatype***

- The datatype of Data Public type tables and their columns of type LONGVARCHAR can be set to VARCHAR(n) at the physical schema level.

**NOTE:** Do not change the datatype or width from Siebel Tools in the logical repository and apply the changes.

When performing ddlsync, dev2prod or upgrade, please make sure that rebuild statements are removed from the schema.sql file.

You do not need to modify the datatype of Repository or System type tables.

## **Using Character Large Objects (CLOBs)**

Siebel objects defined as LONGVARCHARs can also be stored as character large objects (CLOBs) on DB2 UDB for z/OS and OS/390. Siebel eBusiness Applications support CLOBs that allow up to 32 KB of data for a single data element in a table row.

The default setting for these objects in the Siebel Schema is LONGVARCHAR, because of the performance and storage characteristics of CLOBs in a DB2 UDB for z/OS and OS/390 environment. However, as an exception to this rule, customers running Siebel Marketing should change the LONG VARCHAR column in the S\_NOTE table to a CLOB.

The Siebel installation, upgrade, and migration processes create all auxiliary objects necessary to support CLOBs, so that their use is transparent to the user. For more information on the use of CLOBs, see *Upgrade Guide for DB2 UDB for z/OS and OS/390* for your operating system.

**NOTE:** CLOBs are turned off by default. They must be turned on for each table by setting `CI obs = Yes` in the storage control file.

If CLOBs are chosen for a table, the Siebel installation, upgrade, and migration processes create all auxiliary objects necessary to support CLOBs, so their use is transparent to the user.

**NOTE:** If you write your own scripts, it is recommended that you convert columns that contain scripts from LONGVARCHAR to CLOB to safeguard against space limitations that are inherent to the structure of z/OS and OS/390.

For more information on the use of LONGVARCHARs and CLOBs, refer to the *Upgrade Guide for DB2 UDB for z/OS and OS/390* for your operating system.

## **Converting Nonpartitioned Tables to Partitioned Tables**

Siebel eBusiness Applications support two scenarios for converting nonpartitioned tables to partitioned tables.

**NOTE:** In the procedures below, source table refers to your current nonpartitioned table, and target table refers to the destination partitioned table.

- **Source table and target table are defined in different table spaces.** If the source table and target table are defined *in different table spaces*, and the target table space is a new table space that does not yet exist in your current database, no special action is required. Run the two utilities available in the Siebel Software Configuration Utility (Migrate Development to Production, Synchronize Database Schema Definition) to automatically rebuild the table in the new partitioned table space and drop the original table.
- **Source table and target table are defined in the same table space.** If the source table and target table are defined *in the same table space* (for example, if you used the dbconfig.xls spreadsheet to convert an existing table space from nonpartitioned to partitioned), then you must perform one of the following procedures (as appropriate for your configuration) to convert a nonpartitioned table to a partitioned table on the same table space.

**NOTE:** You must perform one of the following procedures, because the database operations do not support rebuilding a table space from nonpartitioned to partitioned. (This results in a validation failure and a halted operation.)

#### ***To convert to a nonpartitioned table for a partitioned table space configuration***

- 1 Use your preferred database tool to manually move every nonpartitioned table (every table that you intend to convert) to a temporary table space, and then drop their original table spaces.
- 2 Use the dbconfig.xls utility to modify the storage control file, changing the mode of the table spaces to partitioned.
- 3 Run the Siebel Software Configuration Utility: Migrate Development to Production. This process automatically rebuilds the source tables in the new partitioned table space and drops the original nonpartitioned table.
- 4 Manually drop the temporary table space that you created in [Step 1](#).

#### ***To convert nonpartitioned tables to partitioned tables using alternate tools***

- Your DBA can use alternate tools (for example, BMC Change Manager or IBM Compare Utility) to convert nonpartitioned tables to partitioned table spaces. In this case, after changes have been applied, the last step is to change the schema version by executing the following SQL:

```
update S_APP_VER
```

```
set CUSTOM_SCHEMA_VER = char(integer(CUSTOM_SCHEMA_VER) + 1)
```

This SQL is stored in ddlview.sql and can be executed outside of the Siebel application.

## Creating Custom Extensions to the Siebel Schema

There are several different scenarios for Siebel schema extensions:

- Creating a small extension column, so the table fits into its existing table space
- Creating a large extension column, so the table has to be redefined to a larger page size and therefore a larger table space

#### ■ Creating an extension table

**CAUTION:** If a LONGVARCHAR column is defined for a table, attempting to add an extension column to the table causes an error. This is because a LONGVARCHAR column uses all the space in a table space that is not used by the non-LONGVARCHAR columns, leaving no space available for an extension column. In this case, use an extension table instead. For further information, see “LONG Columns on DB2 UDB for z/OS and OS/390 and Other DBMS Platforms” on page 142.

The process of creating a small extension column is the same on the z/Series as on other platforms and is documented in *Using Siebel Tools*.

To create a large extension column or an extension table, you must specify a 16-KB or 32-KB table space:

- If developing on DB2 UDB for Windows or UNIX, you must enter a 16-KB or 32-KB table space in the Apply Schema screen. (See [Step 4 on page 149](#).)
- If developing on DB2 UDB for z/OS and OS/390, your DBA must edit the storage control file to specify in which database and table spaces the new extension column or table should reside.

**NOTE:** Creating tables with Identity type columns or extension columns of Identity type is not supported

For information on estimating storage needs, see [Chapter 4, “Preparing for Implementation on the DB2 Host.”](#)

## About Schema Extension Tasks

The following tasks must be performed to extend the Siebel schema:

- Check out and lock the project to which the table being extended belongs. In most cases this is Newtable.

If you want to import data into the new column using Enterprise Integration Manager (EIM), you must check out the project to which the appropriate interface table belongs. For more information, see the sections on EIM in *Using Siebel Tools*.

- Update the logical schema definition in the development environment using Siebel Tools. This works as on other platforms and is documented in *Using Siebel Tools* and *Developing and Deploying Siebel eBusiness Applications*.

**NOTE:** Make sure that you have selected the check box on the Database tab of the Development Tools Options dialog under the View menu. This validates the objects' compliance with DB2 UDB for z/OS and OS/390 and sizing conventions.

- Apply the physical schema extensions to the development database:
  - If developing on DB2 UDB for Windows or UNIX, follow the procedures in [“Applying Schema Extensions to the Development Database” on page 148](#).

- If developing on DB2 UDB for z/OS and OS/390, follow the procedures in [“Migrating Customizations from Development to the Target Database” on page 151](#).

In some companies, direct database extension is not allowed. Siebel Tools does not allow the Apply button to be disabled. The Apply process will fail if developers do not have proper database privileges (such as CREATEDBA and DBADM). For information about the Apply process, see [“Applying Schema Changes to the Target Database” on page 155](#).

If you do not want developers to extend tables, you can control this using database privileges. In this case, developers follow the procedures in [“Migrating Customizations from Development to the Target Database” on page 151](#) to apply schema extensions to their development databases.

- Update and test configuration changes that apply to the extensions.

See *Using Siebel Tools, Developing and Deploying Siebel eBusiness Applications, Applications Administration Guide*, and other documentation on the *Siebel Bookshelf* CD-ROM.

## Creating Storage Control File Objects for New Schema Objects

If you define a new table in Siebel Tools (for example, if you add a new extension table), you must add the table object definition to your storage control file in order for the table to be placed in the appropriate table space.

For example, if you create a new extension table called X\_EXTENT, you must create a corresponding object definition in the storage control file:

```
[Object 6001]
Type=Table
Name=X_EXTENT
Database=SI DB6001
Tablespace=SO600100
CLOBS=NO
```

**NOTE:** If you have different storage control files for your development environment and production environment, then you must add the new table object definition to both storage control files.

## Applying Schema Extensions to the Development Database

Once your customizations are completed, you are ready to update your local DB2 UDB for Windows or UNIX environment.

**NOTE:** Siebel 7 does not support customized database triggers. If you have created customized triggers on your Siebel base tables, you must disable them before updating the logical database schema. You must re-create the triggers after the update is finished.

### To update your local environment

1 In Siebel Tools, select the table from which you want to apply changes to the database.

2 Click Apply in the Object List Editor.

A message box appears, alerting you that you are about to connect to a local database and asking if you want to continue.

3 Click OK.

The Apply Schema message box appears.

4 Fill in the fields as shown in the following table, and then click Apply.

| Field                    | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|--------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Tables                   | <p>Select one of the following options from the drop-down menu:</p> <ul style="list-style-type: none"> <li>■ <b>All.</b> Update the database to reflect all changes made to the dictionary. This option forces each database object to be compared with the data dictionary, and updated if required.</li> <li>■ <b>Current Query.</b> Update the database to reflect modifications made to the tables in the current query only.</li> <li>■ <b>Current Row.</b> Update the database to reflect modifications made to the table in the current row only.</li> </ul> |
| Table space              | Leave blank.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 16K table space          | Leave blank unless you are creating a large extension table that requires a 16-KB table space.                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 32K table space          | Leave blank unless you are creating a large extension table that requires a 32-KB table space.                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| Index space              | Leave blank.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| Table groupings file     | <p>Specify the complete path for the storage control file.</p> <p>This file is provided by the DBA and is specific to your database.</p>                                                                                                                                                                                                                                                                                                                                                                                                                            |
| Privileged user id       | <p>Enter the ID of a database user with CREATEDBA or DBADM privileges.</p> <p>The schema qualifier is read from tool s. cfg.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| Privileged user password | Enter the password of a database user with CREATEDBA or DBADM privileges.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| ODBC data source         | <p>Verify that the ODBC connection specified in the ODBC Data Source text box is correct for your environment.</p> <p>You cannot apply schema changes to any database other than the one you are currently connected to (for example, by specifying the ODBC name of a different database).</p>                                                                                                                                                                                                                                                                     |

5 To activate extensions to EIM tables, select the appropriate tables, and then click Activate.

## How to Spool DDL Generated by the Apply Button

The Siebel environment variable, `SI EBEL_GENERATE_DDL`, allows you to record DDL, generated by clicking the Apply button in Siebel Tools, into a file.

### *To spool DDL, generated by clicking the Apply button, into a file*

- 1 Set the environment variable before running Siebel Tools on Windows by way of the command line or Windows environment parameters:
  - If using the command line, set `SI EBEL_GENERATE_DDL=Y`.
  - If using the Windows environment parameters proceed with the following instructions:
    - a From the Start menu, navigate to Settings > Control Panel > System.
    - b Click the Advanced tab and click Environment Variables.
    - c In the System Variables, click New.  
The New System Variable window appears.
    - d Enter `SI EBEL_GENERATE_DDL` in the Variable name field and `Y` in the Variable value field. Click OK.
    - e Continue to click OK until you are out of the System Variable window.
- 2 Open Siebel Tools and click the Apply button.

The DDL will be generated to the file `output.sql`, which is found in the `$SI EBEL_ROOT/bin` directory, where `$SI EBEL_ROOT` is the directory where Siebel Tools is installed.

## Applying Schema Extensions to the Target Database

Once you have tested your schema customizations in the local environment, you can migrate them to the target user-acceptance or production database. You must perform the following tasks:

- [“Preparing the Target Database”](#)
- [“Migrating Customizations from Development to the Target Database” on page 151](#)
- [“Applying Schema Changes to the Target Database” on page 155](#)

## Preparing the Target Database

Your DBA must complete the following actions before migrating the changes to the target database:

- Prepare the storage control file (for example, `my_storage_file.ctl`).

**NOTE:** If you have created any new extension tables, your DBA must edit the storage control file to specify in which database and table spaces the new extension tables should reside.

- Ask all mobile users to synchronize.
- Make sure all connected clients are disconnected from the Siebel Database Server.
- Once all mobile user transactions have been merged and routed, stop the Siebel Enterprise Server.
- Perform a full backup of the database.

**NOTE:** If you are changing the data type or length of custom extension columns that already have data in the target database, you must export that data before making the schema changes. After making the changes, import the data back into the target database.

## Migrating Customizations from Development to the Target Database

**CAUTION:** Before you migrate your customizations from your development database to your target database, you must validate your storage control file. See [“Validating the Storage Control File”](#) on page 105.

When migrating customizations from your development source to your target database on the DB2 host, you can use one of two modes of execution:

- **Run DDL Automatically.** Select this to apply the DDL and DML required to create the Siebel Schema directly against the database, using an ODBC connection.
- **Generate DDL Into Files.** Select this to generate the DDL required to create the Siebel Schema. You must later use Siebel-provided scripts, your own FTP, or a similar file transfer program to transport the DDL to the zSeries machine, where the DBA executes it using customary methods, for example SPUFI.

This section describes the DDL-generation mode of execution that is generally used under a change-management system.

If your development environment is on DB2 UDB for z/OS and OS/390, your DBA can also clone it to the target database. For more information, see [“Cloning an z/OS and OS/390 Database”](#) on page 158.

**NOTE:** Siebel 7 does not support customized database triggers. If you have created customized triggers on your Siebel base tables, you must disable them before migrating the database schema. You must re-create the triggers after the migration is finished.

### *To migrate the schema under Windows*

- 1 Launch the Database Server configuration wizard:

Choose Start > Programs > Siebel Enterprise Server 7.x.x > Configure DB Server.

or

- a Open a command prompt.
- b Go to the `SI\ESRVR_ROOT\BIN` directory.

- c** Enter the following command:

```
ssi ncfgw -I ENU -v Y
```

- d** In the screen that appears, select `dbsrvr.scm`.

- 2** In the Gateway Name Server Address screen, type the following values as you recorded them in your copy of [Appendix A, "Deployment Planning Worksheet."](#)

- **Gateway Name Server Address.** The alias of the host on which you installed the Siebel Gateway Name Server.

- **Enterprise Server Name.** The name you gave to your Enterprise Server, for example, `si ebel`.

To continue, click Next.

- 3** In the Siebel Server Directory screen, perform either of the following tasks:

- Accept the default value displayed in the Siebel Server Directory field. (This is the `SIEBSVR_ROOT` directory, for example, `D: \sea7xx\si ebsvr.`)

- Use the browse button to select an alternate directory path.

To continue, click Next.

- 4** In the Siebel Database Server Directory screen, perform either of the following tasks:

- Accept the default path displayed in the Siebel Database Server Directory field. (This is the `DBSRVR_ROOT` directory, for example, `D: \sea7xx\dbsrvr.`)

- Use the Browse button to select an alternate directory.

To continue, click Next.

- 5** In the RDBMS Platform screen, select the database platform you use, for example, IBM DB2 UDB for z/OS and OS/390.

To continue, click Next.

- 6** In the Siebel Database Operation screen, select `Migrate Repository`, and then click Next.

- 7** In the ODBC Data Source Name screen, type your source ODBC connection name, and then click Next.

- 8** In the Database User Name screen, enter your source database username and password, confirm the password by typing it again, and then click Next.

- 9** In the Schema Qualifier screen, enter the schema qualifier in uppercase and security group ID from your copy of [Appendix A, "Deployment Planning Worksheet,"](#) and then click Next.

- 10** In the Source Database Repository Name screen, type the source and target database repository names—typically `Siebel Repository` for both, and then click Next.

- 11** In the Target RDBMS Platform screen, select `IBM DB2 UDB for z/OS and OS/390`, and then click Next.

- 12** In the Target Database ODBC Datasource screen, type your target ODBC connection name, and then click Next.



- 13 In the Target Database User Name screen, enter your target database username and password, confirm the password by typing it again, and then click Next.
- 14 In the Target Schema Qualifier screen, type the target schema qualifier in uppercase and the target group authorization ID, then click Next.
- 15 In the Migrate Repository Mechanism screen, select the Generate DDL Into Files option, and then click Next.
- 16 In the DDL Output Directory screen, type the DDL output directory and storage control file, and then click Next.

For example, use *DBSRVR\_ROOT\db2390\dboutput\dev2prod* and *DBSRVR\_ROOT\db2390\my\_storage\_file.ctl*, respectively.

The Configuration Parameter Review screen appears.

- 17 Review the parameters, and then click Finish.

The runnow screen appears, giving you the choice of applying the configuration now or applying it later (and specifying the command line to apply it later).

- 18 Click OK to apply the configuration now.

The Siebel Upgrade Wizard launches.

- 19 Click OK to perform the migration.

The Siebel Upgrade Wizard runs, creating the following files:

- **ftpsync.txt.** FTPs the files that are migrating from development to production on the DB2 Host.
- **jobsync.txt.** Sets up the migration from development to production process on the DB2 Host. The job is sent to the mainframe using *ftpsync.txt*.
- **unload.idc.** Unloads control cards used to unload tables during the migration from development to production process on the DB2 Host.
- **load.idc.** Loads control cards used to load tables during the migration from development to production process on the DB2 Host.
- **schema.sql.** Applies schema changes from the schema ddl data set during the migration from development to production process on the DB2 Host.
- **bumpver.sql.** Bumps up the Siebel Schema version during the migration from development to production process on the DB2 Host.

### ***To migrate the schema under UNIX***

- 1 Source environment variables from `$SIEBEL_ROOT`.

`SIEBEL_ROOT` should be the path of your Siebel installation directory.

`LANGUAGE` should be set to the language in which the Configuration Wizard prompts appear, for example, `enu` for U.S. English.

**NOTE:** `LANGUAGE` is not the same as `LANG`; the former is the language in which the Configuration Wizard runs, while the latter is the code for the Language Pack you installed.

If either of these values is incorrect or empty, reset them using one of the following commands, as appropriate to the shell you use:

#### **csch**

```
setenv SIEBEL_ROOT New Value
```

```
setenv LANGUAGE New Value
```

#### **ksh**

```
SIEBEL_ROOT=New Value ; export SIEBEL_ROOT
```

```
LANGUAGE=New Value; export LANGUAGE
```

- 2 Go to `$SIEBEL_ROOT/bin` and enter:

```
dbsrvr_config.ksh
```

This launches the Database Server configuration wizard.

- 3 Review the values of the following environment variables and confirm whether the settings are correct by entering either Y or N.

**NOTE:** If either the `SIEBEL_ROOT` or `LANGUAGE` value is not set or is incorrect, you must correct it before proceeding.

- 4 Specify the path of your Siebel Server root directory, or accept the default by pressing ENTER.
- 5 Specify the path of your database server root directory, or accept the default by pressing ENTER.
- 6 Enter the number that corresponds to your database platform.
- 7 From the Select Database Operation menu, choose Migrate Repository (4).
- 8 Enter the number that corresponds to the language you are using, typically 1 for English.
- 9 Enter your source ODBC connection name.
- 10 Enter your source database username and password.
- 11 Enter the schema qualifier in uppercase from your copy of [Appendix A, "Deployment Planning Worksheet."](#)
- 12 Specify the source database repository name, or accept the default by pressing ENTER.
- 13 Specify the target database repository name, or accept the default by pressing ENTER.

- 14 Select the target database platform by entering the number that corresponds to IBM DB2 UDB for z/OS and OS/390.
- 15 Enter your target ODBC connection name.
- 16 Enter your target database username and password.
- 17 Enter the target schema qualifier in uppercase.
- 18 Enter the target group authorization ID, or accept the default ID, SSEROLE by simply pressing ENTER.
- 19 In the Select Installation Mechanism screen, select Generate DDL Into Files (1).
- 20 Specify the DDL output directory, for example  
*DBSRVR\_ROOT/db2390/dboutput/dev2prod.*
- 21 Specify the storage control file, for example  
*DBSRVR\_ROOT/db2390/my\_storage\_file.ctl.*  
  
The configuration parameters are listed and you are asked if you want to run `srvrupgwi z`.
- 22 Enter Y for yes.  
  
The Siebel Upgrade Wizard launches.
- 23 You will receive a message that states that the files with the Siebel Schema modifications have been generated in the DDL Output Directory. Select Yes to exit now and apply the files.  
  
The following files are created:
  - **ftpsync.txt.** FTPs the files that are migrating from development to production on the DB2 Host.
  - **jobsync.txt.** Sets up the migration from development to production process on the DB2Host. The job is FTPd to the mainframe using `ftpsync. txt`.
  - **unload.ldc.** Unloads control cards used to unload tables during the migration from development to production process on the DB2 Host.
  - **load.ldc.** Loads control cards used to load tables during the migration from development to production process on the DB2 Host.
  - **schema.sql.** Applies schema changes from the schema ddl data set during the migration from development to production process on the DB2 Host.
  - **bumpver.sql.** Bumps up the Siebel Schema version during the migration from development to production process on the DB2 Host.

## Applying Schema Changes to the Target Database

After generating DDL into files, you must later use Siebel-provided scripts, your own FTP, or a similar file transfer program to transport the DDL files to the zSeries machine, where the DBA executes it using customary methods, for example, SPUFI.

***To apply schema changes to the target Database using Siebel Systems-provided scripts***

- 1** Navigate to the output directory (DBSRVR\_ROOT\DB2390) and open the ftpsync.txt file.
- 2** Edit the ftpsync.txt file using the following information:
  - a** Change &IP into the IP address of your DB2 host, for example ZM01.
  - b** Change &Username into the your own user name, for example SADMI N.
  - c** Change SI EBELQ1 to your own high-level qualifier (HLQ), for example, SADMI N, using a command such as Replace All. Save the file.

Following is an example of the ftpsync.txt file:

```
open &IP
user &Username

quote site cylinders primary=1 secondary=1
quote site recfm=fb lrecl=80 blksiz=0

quote site cylinders primary=1 secondary=1
send &directoryPath1/jobsync.txt 'HLQ.SI EBEL.V00'

quote site cylinders primary=10 secondary=2
send &directoryPath1/unload.lcl 'HLQ.SI EBEL.V01'

quote site cylinders primary=10 secondary=2
send &directoryPath1/load.lcl 'HLQ.SI EBEL.V02'

quote site cylinders primary=5 secondary=2
send &directoryPath1/schema.sql 'HLQ.SI EBEL.V03'

quote site cylinders primary=1 secondary=1
send &directoryPath1/bumpver.sql 'HLQ.SI EBEL.V04'
```

- 3** Execute ftpsync.txt and send the file to the host.

From the command line enter:

```
ftp -vn < ftpsync.txt > ftpsync.log
```

You will be prompted for the password for the user name, which was entered in ftpsync.txt.

Press any key to continue.

- 4** The log file ftpsync.log is created in the DBSRVR\_ROOT\DB2390 directory.  
The log contains information on the file transfer, for example.
- 5** After the transfer is successful, log on to the mainframe, and edit HLQ.SI EBEL.V00D2P.  
Change the job card and HLQ and submit job.  
Make sure that the return code is 0.
- 6** A new PDS is created:  
HLQ.SI EBEL.D2P.EXEC

- 7 Execute the following REXX exec to apply the DDL generated for the schema migration process.  
Execute `clist HLO.SIEBEL.D2P.EXEC(SBLD2P)`
- 8 Verify the schema changes.

## Synchronizing Schema Definitions with the Existing Siebel Schema

The Synchronize Schema Definitions process is used to synchronize the repository definitions with the existing Siebel Schema database. Detailed information about this utility is provided in *Using Siebel Tools*.

On DB2 UDB for z/OS and OS/390 this process depends on the existing database physical layout and the contents of the storage control file. The resulting database storage layout will fully reflect storage control file settings.

When you synchronize repository definitions with the existing Siebel Schema, the following data sources are accessed:

- Siebel Repository
- Storage control file
- DB2 catalog

### To synchronize schema definitions

- 1 Launch the Database Server configuration wizard:  
Choose Start > Programs > Siebel Enterprise Server 7.x.x > Configure DB Server.  
or
  - a Open a command prompt.
  - b Go to the `SIEBSRV_ROOT\BIN` directory.
  - c Run the following command:  
`ssi ncfgw -I ENU -v Y`
  - d In the screen that appears, select `dbsrvr.scm`.
- 2 Select the following options: Run Database utilities > Synchronize Schema Definitions > Generate DDL into files.  
When you generate DDL into files, two output files are generated in the default directory (`$SIEBEL_ROOT\dbsrvr\DB2390\DBOUTPUT\DDLSYNC`):
  - `schema.sql`
  - `ddlview.sql`
- 3 Ask your DBA to apply the output files.

## Cloning an z/OS and OS/390 Database

Your DBA can clone your existing Siebel database to the target database. The DB2 system cloning process involves the following tasks:

- Extracting a storage control file from the existing Siebel schema.
- Using the newly extracted storage control file to generate a DDL file that your DBA will apply to create a clone of the Siebel schema.

To do this, you must run the Database Server configuration wizard twice.

### Extracting a Storage Control File from the Development Database

The first run of the Database Server configuration wizard “reverse engineers” the existing Siebel schema by creating a storage control file (.ctl) file from the database.

For instructions on how to use this utility, see [“Extracting a Storage Control File from the DB2 Catalog” on page 102](#).

### Generating a DDL File from a Storage Control File

The second run of the Database Server configuration wizard uses the newly created storage control file to generate a DDL file. This DDL file will be applied to the database by your DBA to create the Siebel schema clone.

#### *To generate the DDL file under Windows*

- 1 Launch the Database Server configuration wizard in either of two ways:  
Choose Start > Programs > Siebel Enterprise Server 7.x.x > Database Server configuration  
or  
  - a Open a command prompt.
  - b Go to the `SI EBSRVR_ROOT\BIN` directory.
  - c Run the following command:  
`ssi ncfgw -I ENU -v Y`
  - d In the screen that appears, select `dbsrvr.scm`.
- 2 In the Gateway Name Server Address screen, type the following values as you recorded them in your copy of [Appendix A, “Deployment Planning Worksheet.”](#)
  - **Gateway Name Server Address.** The alias of the host on which you installed the Siebel Gateway Name Server.
  - **Enterprise Server Name.** The name you gave to your Enterprise Server, for example, `si ebel`.

To continue, click Next.
- 3 In the Siebel Server Directory screen, perform either of the following tasks:

- Accept the default value displayed in the Siebel Server Directory field. (This is the *SI EBSRVR\_ROOT* directory, for example, D: \sea7xx\si ebsrvr.)

- Use the browse button to select an alternate directory path.

To continue, click Next.

- 4 In the Siebel Database Server Directory screen, perform either of the following tasks:

- Accept the default path displayed in the Siebel Database Server Directory field. (This is the *DBSRVR\_ROOT* directory, for example, D: \sea7xx\dbsrvr.)

- Use the Browse button to select an alternate directory.

To continue, click Next.

- 5 In the RDBMS Platform screen, select IBM DB2 UDB for zSeries and OS/390.

To continue, click Next.

- 6 In the Siebel Database Operation screen, select Install Database, and then click Next.

- 7 In the Select Installation Operation screen, select Install Siebel Database, and then click Next.

- 8 In the ODBC Data Source Name screen, type your source ODBC connection name, and then click Next.

- 9 In the Database User Name screen, type your database username and password, confirm the password by typing it again, and then click Next.

- 10 In the Schema Qualifier screen, type the schema qualifier in uppercase and security group ID for your new database, and then click Next.

- 11 In the Select Installation Type screen, select the Customized Install, and then click Next.

- 12 In the Select Installation Mechanism screen, select the Generate DDL Into Files, and then click Next.

- 13 In the DDL Output Directory screen, type the following, and then click Next.

- DDL output directory, for example: *DBSRVR\_ROOT\DB2390\dboutput\instal*

- new storage control file, for example: *DBSRVR\_ROOT\DB2390\my\_storage\_file.ctl*

The Configuration Parameter Review screen appears.

- 14 Review the parameters, and then click Finish.

The runnow screen appears, giving you the choice of applying the configuration now or applying it later (and specifying the command line to apply it later).

- 15 Click OK to apply the configuration now.

The Siebel Upgrade Wizard launches.

- 16 Click OK to perform the operation when the following message appears.

Files schema.sql and ddlview.sql with the Siebel Schema modifications have been generated in the DDL Output Directory. Please select Yes to exit now and apply the files

The following SQL files are created:

- schema.sql
- ddlview.sql

**NOTE:** If you want to clone the Siebel schema into more than one database, copy the generated DDL file, replace the schema qualifier in the DDL file, and then save the file under a new name.

### ***To generate the DDL file under UNIX***

- 1** Source environment variables from `$SIEBEL_ROOT`.

`SIEBEL_ROOT` should be the path of your Siebel installation directory.

`LANGUAGE` should be set to the language in which the Configuration Wizard prompts appear, for example, `enu` for U.S. English.

**NOTE:** `LANGUAGE` is not the same as `LANG`; the former is the language in which the Configuration Wizard runs, while the latter is the code for the Language Pack you installed.

If either of these values is incorrect or empty, reset them using one of the following commands, as appropriate to the shell you use:

#### **csch**

```
setenv SIEBEL_ROOT New Value
```

```
setenv LANGUAGE New Value
```

#### **ksh**

```
SIEBEL_ROOT=New Value ; export SIEBEL_ROOT
```

```
LANGUAGE=New Value; export LANGUAGE
```

- 2** Go to `$SIEBEL_ROOT/bin` and enter:

```
dbsrvr_config.ksh
```

This launches the Database Server configuration wizard.

- 3** Review the values of the following environment variables and confirm whether the settings are correct by entering either `Y` or `N`.

**NOTE:** If either the `SIEBEL_ROOT` or `LANGUAGE` value is not set or is incorrect, you must correct it before proceeding.

- 4** Specify the path of your Siebel Server root directory, or accept the default by pressing ENTER.
- 5** Specify the path of your database server root directory, or accept the default by pressing ENTER.
- 6** Enter the number that corresponds to your database platform.
- 7** In the Select Database Operation menu, choose Install Database (1).
- 8** In the Installation Operation screen, select Install Siebel Database (1).
- 9** Enter the number that corresponds to the language you are using, typically 1 for English.
- 10** Enter your source ODBC connection name.



- 11 Enter your database username and password, and then confirm the password by typing it again.
- 12 Enter the schema qualifier for your new database in uppercase.
- 13 Enter the group ID for your new database, or accept the default (SSEROLE) by pressing ENTER.
- 14 In the Installation Type screen, select Customized Install (2).
- 15 In the Installation Mechanism screen, select Generate DDL Into Files (1).
- 16 Specify the DDL output directory, for example  
`DBSRVR_ROOT/DB2390/dboutput/install`.
- 17 Specify the new storage control file, for example  
`DBSRVR_ROOT/DB2390/my_storage_file.ctl`.

The configuration parameters are listed and you are asked if you want to run `srvrupgwiz`.

- 18 Enter Y for yes.

The Siebel Upgrade Wizard launches.

- 19 Enter Y to perform the operation when the following message appears.

Files `schema.sql` and `ddlview.sql` with the Siebel Schema modifications have been generated in the DDL Output Directory. Please select Yes to exit now and apply the files

The following SQL files are created:

- `schema.sql`
- `ddlview.sql`

**NOTE:** If you want to clone the Siebel schema into more than one database, copy the generated DDL file, replace the schema qualifier in the DDL file, and then save the file under a new name.

## Migrating Rules and Other Data

After you have successfully migrated your source schema to your target database, you must copy data from your development environment. Such data might include:

- Modified files, such as the Siebel repository file (`.srf`), Web templates, image files, and cascading style sheets
- Transactional data, such as accounts, contacts, and opportunities
- Setup data, such as employees, positions, and responsibilities
- Program data, specifically Assignment Manager rules and Workflow processes and policies, and personalization rules and expressions

For information on copying such data, see [Chapter 12, “Migrating Data Using Siebel EIM,”](#) and *Developing and Deploying Siebel eBusiness Applications*.

## Applying Schema Changes to Other Local Databases

For information on upgrading local databases, see *Using Siebel Tools* and *Developing and Deploying Siebel eBusiness Applications*.

## Customizing Applications Using Assignment and Workflow Rules

There are two additional ways of customizing applications besides using Siebel Tools.

Siebel Assignment Manager allows you to create business rules that automatically assign entities such as opportunities, service requests, or activities to the most qualified individuals. For more information, see *Siebel Assignment Manager Administration Guide*.

Siebel Workflow provides a graphical interface for designing and implementing business processes and user interactions. Workflow processes define the steps to automate business processes, such as sending email. Workflow policies trigger processes when they detect certain conditions, such as an opportunity being assigned. For more information, see *Siebel Business Process Designer Administration Guide*.

To activate rules—whether they are assignment or workflow—you must run the following Siebel Server components:

- **Generate Triggers (GenTrig).** Allows you to create database triggers.

GenTrig writes database triggers to a SQL file, such as TRIGGER.SQL. On the DB2 host, the SQL file is executed manually by a DBA. For manual execution of TRIGGER.SQL, run GenTrig with the EXEC parameter set to FALSE.

**NOTE:** For DB2, GenTrig does not log into the database as the table owner. Instead, it logs in as the privileged user. All triggers generated are qualified with the schema qualifier. When starting the GenTrig component, users are prompted for the schema qualifier.

- **Workflow Monitor.** Uses these database triggers to identify which records might match policy conditions.

For information on activating rules, see *Developing and Deploying Siebel eBusiness Applications*.

# 11

## Maintenance Considerations for zSeries

This chapter describes maintenance tasks that you can perform to improve performance of Siebel eBusiness Applications on DB2 UDB for z/OS and OS/390 anytime after you complete the Siebel application installation.

This chapter consists of the following topics:

- [“About DB2 Statistics for Siebel eBusiness Applications”](#)
- [“About Reorganizing Table Spaces, Partitions, and Indexes” on page 164](#)
- [“About Cursor Close” on page 165](#)
- [“Setting Database Connection Pooling” on page 170](#)
- [“About Siebel Dynamic SQL” on page 170](#)
- [“Using odbcsql to Submit SQL Statements” on page 173](#)
- [“About DB2 Dynamic Statement Caching” on page 174](#)
- [“About Tracing the Source of a Query” on page 175](#)
- [“About Coordinated Universal Time and DB2 UDB for z/OS and OS/390” on page 176](#)

### About DB2 Statistics for Siebel eBusiness Applications

It is recommended that you apply statistics to Siebel applications. The DBA should update statistics on table spaces that contain EIM interface tables and base tables (including table spaces that contain extension tables and repository tables) when there has been a change of 20 percent or more in the row distribution. It is usually not necessary to update statistics on all of the table spaces, only on those containing tables that have changed.

RUNSTATS and DSTATS utilities are recommended for updating statistics.

#### About the RUNSTATS Utility

Execution of the rstat390 utility from midtier executes RUNSTATS on the DB2 host. The rstat390 script is located within the *SIEBSRVR\_ROOT/bin* directory. For rstat390 help, use option /h.

You should update statistics only when there is little activity on the system, such as after midnight. If you run this utility while users are accessing and updating the Siebel database, lock contention can occur. When this happens, an error message is generated. As an example, the following error message is generated from rstat390.

```
ODBC error S1000 in SQLExecDirect: [IBM][CLI Driver][DB2/6000] SQL2310N The utility could not generate statistics. Error "-911" was returned.
```

This does not harm your database, but the RUNSTATS job will have to be rerun for any table for which this type of error was generated, because statistics were not updated for that table.

You can execute RUNSTATS on an active system if you specify `shrlevel change` as an option. This allows concurrent access while the RUNSTATS utility executes.

You should execute RUNSTATS following reorganization of any fragmented table spaces and indexes. For more information on this topic, see [“About Reorganizing Table Spaces, Partitions, and Indexes.”](#)

You can also execute RUNSTATS directly from the DB2 host.

### **About the Distribution Statistics Collection Program (DSTATS) Utility**

RUNSTATS collects distribution statistics, but only in certain cases: for example, only for columns which form the senior key of an index, and only for the most (rather than least) frequently occurring values.

A slow query may be a result of an inefficient access path chosen by the optimizer due to data skew on a non-leading indexed column. In such cases, IBM recommends running the DSTATS utility to collect column distribution statistics on such columns. Running DSTATS could significantly improve the query time. For example, on a query with search predicate `PRIV_FLG = 'N'`, Siebel Systems has seen an improvement from 0.5 seconds to 0.04 seconds with column distribution statistics collected on `S_CONTACT.PRIV_FLG`.

For more information on DSTATS and to access the download from IBM, see Siebel SupportWeb.

## **About Reorganizing Table Spaces, Partitions, and Indexes**

It is recommended that you reorganize tables spaces, partitions, and indexes that have a tendency to become fragmented.

No strict guidelines can be offered as to which table spaces, partitions, and indexes might require reorganization due to the variety in application and customer operation variables at any given customer site. However, database administrators should pay attention to the status of large or heavily used table spaces and indexes, because fragmentation of these objects can affect performance significantly. (For a list of suggested Siebel tables to monitor for reorganization, see [Table 8 on page 41.](#))

You should avoid using online REORG of Siebel repository table spaces. Reorganization of Siebel repository table spaces should be scheduled during application downtime to avoid adversely affecting the Siebel application.

After you reorganize table spaces and indexes, it is recommended that you execute `rstat390` (from `midtier`) or RUNSTATS (on the DB2 host) . For more information on how to run these utilities, see [“About the RUNSTATS Utility” on page 163.](#)

## About Cursor Close

Cursor close allows you to regulate system resource utilization by Siebel clients in an z/OS and OS/390 environment by setting cursor configuration parameters.

The cursor configuration parameters specify the number of database rows a Siebel client cursor can get from a user-entered query or a Siebel operation that generates a SQL query.

The cursor configuration parameters, MaxCursorSize and PreFetchSize, are read at startup from the following places:

- Enterprise profile configuration parameters on the Siebel Server for Siebel Web Clients.
- Application configuration (.cfg) file for Siebel Dedicated Web Clients and Siebel Mobile Web Clients.

The Siebel client connector closes the SQL cursor after the number of rows specified by cursor close parameters DSMaxCursorSize and DSPreFetchSize have been retrieved. By default on other DBMS platforms, Siebel Systems sets DSMaxCursorSize and DSPreFetchSize to values of -1 and -1 which causes the cursors to remain open to retrieve additional data.

When using DB2 for z/OS and OS/390 v7, setting the values of the DSMaxCursorSize and the DSPreFetchSize parameters to -1 can cause a high number of DB2 Database Active Threads (DBAT); this can potentially lead to DB2 DBM1 address space task abends (abnormal ends) caused by DBM1 virtual storage constraints. To alleviate this problem, Siebel Systems has recommended setting the MaxCursorSize and PreFetchSize for zSeries customers to positive values. The default value for both MaxCursorSize and PreFetchSize is 128. This allows a reasonable number of rows to be retrieved without holding open the cursor. Once the cursor is closed, the change is committed and the Database Active Thread (DBAT) becomes inactive.

Siebel Systems recommends that your corporate IT department and Siebel administrator collaborate to determine the values for these parameters that are most appropriate for each class of users connecting to DB2 UDB for z/OS and OS/390. These guidelines are not mandated, but provide a balance between system resource utilization, user responsiveness, and productivity. Your enterprise might have unique requirements beyond the scope of this chapter. For assistance, contact Siebel Expert Services.

## About Cursor Operation in Unrestricted and Restricted Modes

Cursor operation can be in either unrestricted or restricted mode.

### Unrestricted Mode

When cursor configuration parameters MaxCursorSize and PreFetchSize are not specified or are set to a value of -1, the Siebel application opens a cursor on the user's behalf and retrieves blocks of records until the query result set is exhausted or until the user cancels the display of remaining records. When a user remains on a list applet with multiple screens of data, for example Contact List Applet, the application continues to hold the cursor.

The impact on database and network resources when a query returns a large result set can be significant.

## Restricted Mode

During normal operation (restricted mode), with cursor configuration parameters set appropriately, the Siebel application retrieves up to the MaxCursorSize number of rows from the DB2 database.

For example, if MaxCursorSize were specified as 128, the Siebel application would retrieve up to 128 rows from the database for a single operation or query. The Siebel application would then close the cursor, releasing the thread for other users to access. The user can then scroll through the retrieved rows, because the rows are cached on the Siebel Server within the Application Object Manager.

If a user tries to scroll to records beyond the value of MaxCursorSize (128 in the example), the cursor close alert appears:

There were more rows than could be returned. Please refine your query to bring back fewer rows.

For example, let us say that the user navigates to the My Opportunities view, which is sorted by the Name column. Seventy-five rows qualify for this query. The initial query returns 128 rows; the 128th row says Siebel Systems. When the user tries to scroll to the 129th opportunity, the cursor close alert appears. The user would then have to choose Edit > Query > Refine Query, type '>Siebel Systems' in the Name field, and then execute the query to retrieve additional records.

## Modifying the Cursor Close Alert Message

The message that is displayed to the user when there are more rows than can be returned can be modified. This is performed in the System Preferences view under the Application Administration screen.

### *To modify the cursor close alert message*

- 1 In your Siebel application, navigate to Site Map > Application Administration > System Preferences.  
The System Preferences view appears.
- 2 In the System Preference Name column, select SSASql ErrRsltDiscarded.
- 3 Edit the value in the System Preference Value column to display your message.  
The maximum message length is 200 characters.
- 4 Restart the Siebel application.

## Cursor Close Ignored

Many specialized Siebel business components, such as Application Administration, List Management, all charts, all picklists (List Of Values), and all calendar views have been modified to ignore the cursor configuration parameter settings. Ignoring the settings allows these functions to operate correctly.

## About Cursor Configuration Parameter Values

You can change cursor configuration parameters for Siebel clients in the following locations:

- Application configuration (.cfg) files for the Siebel Dedicated Web Client and Siebel Mobile Web Client.
- From the command line for the Siebel Web Client.  
Because the Siebel Web Client does not have an application configuration file, it uses the configuration parameter values specified for the Enterprise Server; these values are applied to all server components and the Web Client.
- Application configuration (.cfg) file for Siebel Tools.
- At the business component level in Siebel Tools.

For information on Siebel clients, see the *Siebel Installation Guide* for the operating system you are using. For more information on Siebel Tools, see the *Using Siebel Tools* guide.

## Setting Cursor Configuration Parameter Values for the Siebel Web Client and Siebel Servers

For the Siebel Web Client, cursor close parameter values are set from the command line at the Siebel Enterprise Server level. As a result, you cannot set them separately for each class of users. You must set them to the same value for all users, typically 128.

For more information on modifying enterprise server parameters, see the *Siebel System Administration Guide*.

### ***To set cursor configuration parameter values for the Siebel Web Client***

- 1 List the cursor configuration parameters to check their current value:
  - a From the command line in Windows or UNIX, navigate to the `srvrmgr` prompt.
  - b Enter the following:
 

```
srvrmgr > list advanced param DSMaxCursorSize for named subsystem ServerDataSrc
srvrmgr > list advanced param DSPreFetchSize for named subsystem ServerDataSrc
```
- 2 Change the values of the cursor configuration parameters by entering the following from the command line:

```

srvrmgr > change parameter DSMaxCursorSize=value for named subsystem ServerDataSrc
srvrmgr > change parameter DSPreFetchSize=value for named subsystem ServerDataSrc

```

where *value* is the new value you want to specify for the cursor configuration parameters.

**NOTE:** DSMaxCursorSize and DSPreFetchSize must be set to the same value.

- 3 Stop and restart the Siebel Server for the changes to take effect.

## Setting Cursor Configuration Parameter Values for the Siebel Dedicated Web Client

For the Siebel Dedicated Web Client, cursor configuration parameter values are set in application configuration files. They are set based on user classes. Table 17 lists the user classes and the recommended cursor configuration parameter values for each class.

Table 17. User Classes and Recommended Cursor Close Parameter Values

| User Class                   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Recommended Value |
|------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|
| Siebel administrator         | Accesses the system to perform system updates and might need to access large amounts of data.                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | -1                |
| Siebel application developer | Performs development and customization of the Siebel application using Siebel Tools.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | -1                |
| Special                      | <p>Sporadically submits queries or executes operations returning a large number of rows. Special users cannot be effective if their queries are limited by the cursor configuration parameters.</p> <p>Examples include:</p> <ul style="list-style-type: none"> <li>■ No requery formulation is available to access the next set of data.</li> <li>■ Operation requires the complete result set.</li> <li>■ Data returned must be internally consistent. When a requery is issued, the returned data set might be changed because of other database activity.</li> </ul> | -1                |
| Typical                      | Majority of users. Examples include call center agents and sales representatives.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | 128               |

For information on application configuration files, see the *Siebel Installation Guide* for the operating system you are using.

### To set cursor configuration parameter values for the Siebel Dedicated Web Client

- 1 Create a backup copy of the application configuration (.cfg) file.



- 2 Open the original .cfg file in a text editor.
- 3 Set the MaxCursorSize and PrefetchSize parameters in the [ServerDataSrc] section of the .cfg file.

**NOTE:** MaxCursorSize and PrefetchSize must be set to identical values.

## Setting Cursor Configuration Parameter Values for the Siebel Mobile Web Client

For the Siebel Mobile Web Client, cursor configuration parameter values are set in application configuration files. Determine the cursor configuration parameter values based on the usage and network access of the Siebel Mobile Web Client user:

- For users who connect exclusively to the local database and update by synchronizing with the corporate Siebel Database Server, no cursor configuration values need to be set. These users do not have cursor configuration parameters in the [Local] sections of their .cfg files.
- For users who access their local database and connect to the corporate Siebel Database Server, set values based on the type of user. See [Table 17 on page 168](#).

Siebel administrators must take special care with the training of Siebel Mobile Web Client users who access both their local database and the Siebel Database Server. This training must include explaining the difference in operation between the two platforms, setting appropriate expectations for receiving cursor close alerts when connected to the server database, and showing how to formulate queries for common cursor close alert scenarios.

For information on application configuration files, see the *Siebel Installation Guide* for the operating system you are using.

### *To set cursor configuration parameter values for the Siebel Mobile Web Client*

- 1 Create a backup copy of the application configuration file.
- 2 Open the original .cfg file in a text editor.
- 3 Set the MaxCursorSize and PrefetchSize parameters in the [ServerDataSrc] section of the .cfg file.

**NOTE:** MaxCursorSize and PrefetchSize must be set to identical values.

## Using Siebel Tools to Bypass the Number of Rows Returned

There may be cases in which you do not want to change the general limitation concerning the number of rows returned (default 128), but instead want to have unlimited number of rows available for certain operations.

To set this up, you would leave the parameters MaxCursorSize and PrefetchSize in the .cfg file set to the default number, or any other number, but use Siebel Tools to set the affected Business Components related to unrestricted operation to bypass the limitation.

To ignore the Cursor Close limitation, you would change the properties for parameters MaxCursorSize and PrefetchSize for these Business Component to -1.

## Setting Database Connection Pooling

Siebel 7 introduced Siebel database connection pooling which decreases the necessary number of DB2 DBATs. The use of Siebel database connection pooling may allow Siebel zSeries customers to use the default DSMaxCursorSize=-1 and DSPreFetchSize=-1 with careful DB2 buffer pool tuning and utilizing DB2 storage contraction DSNZPARMs as documented in this guide.

Siebel Object Manager server components offer the ability to share and reuse database connections, a feature that is referred to as Siebel Database Connection Pooling and Multiplexing. This feature saves the overhead of creating new database connections for each user session and allows the number of concurrent user sessions to exceed the number of database connections.

When you enable Siebel Database Connection Pooling and Multiplexing, the Object Manager component maintains a pool of database connections. When a new connection is needed (a user session is started), the Object Manager looks for an unused connection in the pool. If no connections are available and the pool has less than the maximum number (MaxSharedDbConns) of connections, then a new connection is created. When the maximum number of connections is reached, the Object Manager assigns the session to an existing connection and multiplexes the database operations from sessions sharing the connection. When the number of concurrent sessions requires sharing connections, the Object Manager attempts to share the connections equitably. As sessions terminate, the Object Manager keeps track of connection utilization and attempts to maintain the minimum number (MinSharedDbConns) of connections in the pool. If there are unused connections, they are released until there are only the minimum number.

Three parameters control Siebel Database Connection Pooling and Multiplexing:

- **MinSharedDbConns.** Minimum number of shared database connections used for general purpose operations.
- **MaxSharedDbConns.** Maximum number of shared database connections used for general purpose operations.
- **MinTrxDBConns.** Minimum number of dedicated database connections used for special purposes.

For more information on configuring Database Connection Pooling and Multiplexing, see the *Performance Tuning Guide*.

## About Siebel Dynamic SQL

Traditional applications are typically based on static, hard-coded SQL. However, Siebel eBusiness Applications generate dynamic SQL statements at run time, using the Siebel Application Object Manager and the program configuration in use by the customer. In this way, SQL statements can be as individual as the deployments in which they occur and, in some cases, dynamic SQL might become so complex, that it pushes the limits of DB2. For example, joins with 30-40 tables are not uncommon.

To reduce the complexity of generated SQL (such as the number of joins, the number of columns in SELECT statements, or the number of tables in FROM statements), consider eliminating columns from the standard Siebel applets that you do not require for your business needs, especially in the most frequently used screens and views. You should also keep this consideration in mind when you customize the Siebel application by creating new objects.

## About Spooling SQL Within the Siebel Application

Spooling with the Siebel application can help in the following ways:

- To enable the SQL statement being executed to be identified when an SQL error code is returned.
- It can be used to assess the time take by the various SQL statements when performance issues are encountered.
- When data appears to be missing from a view, the SQL used to retrieve that data can be identified.

This can then be run manually, either from the midtier machine, for example using the DB2 Command Center, or on the z/OS, for example using tools such as SPUFI, to see exactly what data is being returned.

To minimize the amount of SQL tracing produced, and therefore, the size of the log files, it is best to limit the tests to the minimum number of screen changes needed to allow the issue to be reproduced.

Generally, it is easier to spool SQL on a dedicated client as you will have more control over the testing. This makes it easier to minimize the amount of SQL traced.

## Turning on SQL Spooling for the Object Manager and the Database Connector

Component event logging can be turned on for the Dedicated and Mobile Web Clients and the server component. Within this category, you can spool the SQL generated by the Siebel object manager and by the database connector. You can have either the object manager spooling or the SQL spooling, but not both as the amount of data will be confusing.

### *To turn on SQL spooling for the object manager*

- To turn on SQL spooling for the object manager, set the component event Object Manager SQL Log to 4.

### *To turn on SQL spooling for the database connector*

- To turn on SQL spooling for the database connector, set the component event SQL Parse and Execute to 5.

For some components, the SQL Profiling event can be used to find the worst performing queries.

For dedicated clients, you can also set these component events as environment variables. They work in addition to the /s spooling options.

## Enabling DB2 CLI Tracing

DB2 Call Level Interface (CLI) tracing can be enabled by adding the following lines to the db2cli.ini file.

***To enable DB2 CLI tracing***

- 1** Enter a section into the DB2CLI.INI file called [COMMON].
- 2** Add the following entries to your DB2CLI.INI file.

```
[COMMON]
Trace=1
TracePathName=C: \TRACE
TraceFlush=1
TraceComm=1
```

One trace file will be created for each thread in the path specified. The path must already exist. On UNIX systems, the DB2 process owner must have the appropriate permissions to be able to write into the directory specified in the TracePathName.

The resulting files will be called pid.tid i.e. 382.0 or -23456.0. Only the trace files created at the time of problem recreation should be reviewed.

This section can be added into the db2cli.ini file for both Windows and UNIX clients. Each DB2 process will generate its own trace file. The events in the DB2 CLI trace are specific to DB2 clients.

**Enabling ODBC Tracing**

ODBC tracing can be enabled by following the instructions in FAQ 1343 on Siebel SupportWeb. ODBC tracing produces results similar to the CLI trace but it focuses more on the ODBC API calls. You will not be able to see the data being passed in the SQLExecute calls with the ODBC trace but you can see it in the DB2 CLI traces.

**Enabling SQL Tracing for Database Utilities**

SQL Tracing is enabled by setting this environment variable before running the Siebel database utilities.

***To enable SQL Tracing for database utilities***

- set DBUTIL\_LOG\_EVENTS=SQLParseAndExecute=5, SQLDBUtilityLog=3

**About SQL Tracing on the DB2 Host Using Tools Like DB2 Performance Monitor**

Tracing on the DB2 host should be set in conjunction with the DBA. Refer to the documentation for the performance monitoring tool being used to determine how to enable the tracing.

## About a RLF Error Message

DB2 Resource Limit Facility (RLF) can be set up to cancel long running queries. For more information on using RLF, refer to a corresponding IBM documentation.

When performing long-running queries, your end users receive a RLF error message. The error message reflected is:

An error has occurred executing a query. Query did not execute successfully because it exceeded the resource limits set on the database server.

Please simplify your query or report this problem to your systems administrator.

This error occurs when an SQL statement is canceled by the DB2 Governor because it exceeded the resource limits set on the database server.

If this message is received because of executing a QBE (Query by Example), end users should try to simplify the query. Otherwise, contact their systems administrator regarding what they did with the application that resulted in this error.

## Using odbcsql to Submit SQL Statements

The Siebel Server installation program installs a utility called odbcsql in the *SIEBSRVR\_ROOT\bin* directory of your Siebel Server. Siebel eBusiness Applications use this program to perform operations on the DB2 database whenever these operations are performed from the midtier—that is, from the Siebel Server.

You can use odbcsql to submit SQL statements, for example, to obtain DB2 storage group names for data and indexes that you will be prompted for during the Siebel Database Server installation process.

**NOTE:** You can also use any other utilities that allow you to query and return result sets to the client.

As noted earlier, you can also use odbcsql to test your ODBC data source after you install the Siebel Server.

### To execute odbcsql

- Navigate to the *SIEBSRVR\_ROOT\bin* directory and enter the following text at a command prompt, as appropriate for your operating system:

```
odbcsql /source database_alias /user username /password password
```

for example:

```
odbcsql /s siebsrvr_siebel /u sadmi n /p sadmi n
```

where:

/s = the ODBC DSN for the connection.

/u = the user ID.

/p = the password.

If `odbcsql` connection is successful, you can then enter queries against the DB2 catalog tables, for example:

```
SELECT NAME FROM SYSIBM.SYSTOGRGROUP;
```

This query returns a list of all DB2 storage groups. To exit `odbcsql`, enter `EXIT`.

For additional information, enter the following at a command prompt:

```
odbcsql /hel p
```

The `odbcsql` utility is also useful in troubleshooting postinstallation connectivity problems that come from the ODBC layer of your installation.

## About DB2 Dynamic Statement Caching

Siebel eBusiness Applications take advantage of DB2 global statement caching. This allows dynamic SQL statements that are reexecuted to bypass the `PREPARE` phase after the first execution. Dynamic statement caching allows dynamic statements to be prepared once, stored on a global statement cache, then reused many times. The Siebel application is also designed to maximize the reuse of statement caching.

You enabled global statement caching when you set the `CACHEDYN` parameter to `YES` in macro `DSN6SPRM` when generating `DSNZPARMS`. Another method of enabling global statement caching is to set `Dynamic Cache SQL` to `YES` when installing DB2.

Because DB2 saves the cached statement in the Environmental Descriptor Manager (EDM) pool, proper sizing of this pool is very important. For guidance on sizing the EDM pool, see your IBM documentation.

**NOTE:** The EDM pool should be monitored and adjusted based on your workload.

Other system parameters associated with dynamic statement caching are illustrated in [Table 18](#).

Table 18. System Parameters Associated with Dynamic Statement Caching

| Parameter | Value  | Description                                                            |
|-----------|--------|------------------------------------------------------------------------|
| EDMDSPAC  | 10,000 | Overrides size of data space for EDM pool during statement caching.    |
| EDMPOOL   | 32     | EDM pool storage size.                                                 |
| CONSTOR   | YES    | Contracts the EDM pool after 50 commits or >2 MB in use by one thread. |

### How to Invalidate Cached Statements

After DB2 prepares a SQL statement and places it in the global statement cache, it gets reused. If a customer is performing SQL tuning and it is desirable to force DB2 to prepare and reoptimize a cached statement, the global SQL statement cache must be invalidated.

To invalidate an entry in the global statement cache for a specific SQL statement, one of the tables referenced by the SQL statement or the table catalog statistics must be altered in some way.

Methods of invalidating a dynamically cached statement include the following:

- Use the RUNSTATS utility to automatically invalidate cached statements that refer to objects against which RUNSTATS was executed. The RUNSTATS utility can be run on any table or table space referenced in the SQL statement.
- Use the DB2 command `STOP OBJECT` or the SQL `DROP`, `ALTER`, or `REVOKE` statements (for example, `REVOKE ALL FROM PUBLIC`) on any object related to the plan to invalidate both global and local cache. While some `ALTER` statements might require an object to be stopped, others might not; for example, `STOP OBJECT` is not required for `DROP` or `REVOKE`.
- Use the command `ALTER TABLE tablename AUDIT NONE` to purge statements that reference that table from the cache.
- Use a `CREATE INDEX` statement on tables used in the DB2 access path.
- Stop the DB2 subsystem and restart it.

SQL generated by the Siebel application will be sent to the DB2 host on the z/OS platform through ODBC and the DB2 Call Level Interface (CLI) before the SQL command is processed by DB2. To effectively troubleshoot issues it may be necessary to trace the SQL statement through one or all of these layers. This should normally be done under the guidance of Siebel support personnel who will advise on the most appropriate tracing for the situation. Steps to initiate the tracing at each layer are described below.

## About Tracing the Source of a Query

Using Session ID, administrators can identify from which terminals queries are generated.

The session ID is composed of three parts: host IP in decimal, process ID, and serial number. A session ID might look like this: 172.20.80.25\_324\_1.

Eventually the object manager and DB2 host would share the same string that can be used to identify the current connection.

Once the session ID is displayed at the Server Manager, you can also check it on the database side.

### To verify session ID on DB2 host

- 1 At the Option prompt, enter the location of the DB2 Primary Option Menu.  
The DB2 Primary Option Menu appears.
- 2 At the Command prompt, enter D to access the DB2 Defaults Panel.
- 3 At the Command prompt, enter the name of the subsystem that you are connecting to.
- 4 Press F3 to go back to the DB2 Primary Option Menu.
- 5 At the Command prompt, enter the selection that will take you to the DB2 Command window.
- 6 At the DB2 Command window, enter the following:  
-DI SPLAY THREAD(\*) DETAIL

**NOTE:** The "-" is required.

**7** Press ENTER to display current active threads.

The session ID string is displayed in the application name field. In case there are many active threads shown here, you should be able to find out your thread by matching the corresponding host name and userid.

The session ID is composed of *client IP\_client process id\_serial #*.

## About Coordinated Universal Time and DB2 UDB for z/OS and OS/390

The *Global Deployment Guide* contains information on using Coordinated Universal Time (UTC). However, when installing and configuring UTC on DB2 UDB for z/OS and OS/390, you should consider the following:

- The UTC upgrader is not supported on DB2 for z/OS and OS/390.

You should configure UTC using a fresh database install only.

- The Siebel UTC implementation will only work if the system time that the database software is using is in UTC.

Any system or database parameter that implements a timezone other than UTC must remain at zero. The use of any other value is not supported, as it effectively returns the database configuration to a local timezone.

As an example, the z/OS or OS/390 system parameter called TIMEZONE can be set in the CLOCKxx member of SYS1.PARMLIB. For a correct Siebel UTC implementation, the value of this parameter must be zero, as reflected here:

```
TIMEZONE W. 00. 00. 00 /* GMT */
```



# 12 Migrating Data Using Siebel EIM

This chapter describes special considerations for customers using Siebel Enterprise Integration Manager (EIM) to import, export, update, merge, or delete data within Siebel eBusiness Applications on DB2 UDB for z/OS and OS/390.

Before using EIM, familiarize yourself with the *Siebel Enterprise Integration Manager Administration Guide*. This chapter only discusses those elements of EIM operation specific to running EIM on DB2.

This chapter consists of the following topics:

- [“Setting Up EIM for DB2”](#)
- [“Improving EIM Performance During the Import Process” on page 180](#)
- [“Considerations for EIM Tables” on page 183](#)

## Setting Up EIM for DB2

There are many ways to optimize EIM data throughput. EIM is similar in function to other DB2 applications, so your DBA should use regular performance monitoring tools to monitor and fine-tune performance.

You should review and tune performance for each set of business data you load with EIM, for example, Opportunities, Contacts, or Products.

## Choosing an Appropriate Partitioning Scheme

To achieve the best system throughput, partition base tables and EIM tables and set up parallel EIM processes to efficiently exploit partitioned EIM table spaces.

When a table is created on a partitioned table space, the table is assigned a partitioning key composed of one or more columns. EIM tables by default use a clustering U1 index in the order of IF\_ROW\_BATCH\_NUM, ROW\_ID. In some cases, you can obtain better performance by partitioning EIM tables based on the partitioning key of the target base table, for example, on IF\_ROW\_BATCH\_NUM plus the partitioning key of the corresponding target table.

**NOTE:** The values in this key should not be updated. Changes to values in the partitioning key might require unloading and reloading or performing a REORG on the data to resolve performance issues.

For more information on DB2 partitioning, see [Chapter 6, “Understanding Siebel Partitioning.”](#)

## Considering the Effect of EIM on Table Partitioning

The mechanism by which EIM generates the ROW\_ID can result in an uneven distribution of data if you use EIM with one of the default partitioning schemes to import data into a base table. See [“About EIM Considerations for Table Space Partitioning” on page 65](#) for recommendations about how to avoid uneven distribution of data in a partitioned table that has data imported by EIM.

## Planning Data Mapping

Data mapping—an important part of preparing to run EIM—is particularly important on DB2 due to the way DB2 stores data in tables and indexes. Your data mapping should take into account such factors as partitioning, look-ups during the load phase, and searches for online transactions.

### Importing Data to the Siebel Application for DB2

The Siebel eBusiness Applications Data Model provides for many possible business needs and configurations. Many of the tables, columns, and indexes provided with Siebel eBusiness Applications might not be relevant to your business. Because DB2 stores information in all columns, including NULL columns, you should carefully consider which data should be imported into your Siebel application as part of your initial planning. Unnecessary data increases the size of tables and indexes, which can affect performance.

For information on dropping columns and indexes and reducing column sizes using Siebel Tools, see [Chapter 10, “Customizing a Development Environment.”](#)

### Using the MS\_IDENT Column for DB2 UDB for z/OS and OS/390

The MS\_IDENT column exists in every EIM table and is specific to DB2 UDB for z/OS and OS/390. It allows DB2 to automatically generate unique and sequential values for each row in a table.

This column is the unique identity column created using the *Generated Always with Identity* option.

**CAUTION:** If you load data into this column using load utilities, your import might fail. This can particularly be an issue when migrating data from a Siebel application on DB2 for UNIX or Windows to a Siebel application on DB2 UDB for z/OS and OS/390.

### Presorting Data During Initial Load

When loading the EIM tables with legacy data, sort this data on the partitioning key that will be used on the target base table. When sorting string data, use the same character-collating sequence used in the target database. During this load, you should also preassign and load batch numbers corresponding to the partitioning scheme of the target base tables when possible. This is particularly useful for those processes such as Accounts or Contacts, where the partitioning key is based on business data rather than on a generated key. This generally improves EIM throughput, as the data is then loaded sequentially and by partition into the target base table.

**TIP:** To further improve performance, once the EIM tables are loaded with partitioned data, you can unload this data in clustering order, then reload it.

## Choosing an Optimal EIM Batch Size

Users can regulate EIM commit frequency in several ways, including:

- Setting `COMMIT EACH TABLE = TRUE`.
- Setting `COMMIT EACH PASS = TRUE`.
- Adjusting batch size.

Of these, the batch size most directly influences EIM performance.

The size of EIM batches can significantly affect throughput. This occurs as a result of the number of commits that EIM must execute. While each commit is CPU-intensive, commits release page locks, thereby releasing memory and avoiding lock escalation and time-outs.

While every installation is different, you should perform initial testing with large batch sizes (except `DELETE EXACT`, which normally runs better with smaller batch sizes). You might want to use this as a starting point for your own processes and modify batch size based on results.

**NOTE:** Processes that populate child tables might violate the maximum number of locks allowed by the system or cause contention for resources in those child tables. To avoid this, reduce the batch size of these processes.

## Choosing an Optimal Number of Parallel Streams

A rule to determine how many batches can run simultaneously is that the combined number of parallel streams should keep the CPU 100% busy. These can either be multiple EIM processes running simultaneously or the same EIM process repeated against multiple partitions of the same table.

Running EIM processes in parallel on a DB2 database with the default setting of `UPDATE STATISTICS = TRUE` can cause a deadlock when multiple EIM processes access the same interface table simultaneously. To avoid a deadlock, set `UPDATE STATISTICS = FALSE` in the EIM configuration file.

A similar problem might occur if parallel processes access the same child tables while updating foreign keys. Therefore, analyze your EIM mappings and processes and do not run these processes in parallel. Alternatively, you can try altering the locking level from Page to Row.

## Optimizing DB2 Database Logging

The number of parallel streams that the DB2 UDB for z/OS and OS/390 environment can support is often limited more by the DB2 logging throughput rate than anything else. The commit process waits for a successful write to the active log and, if there is contention on the logs, performance suffers. Possible solutions to ease bottlenecks caused by logging performance include:

- Increase the number of output buffers for writing active log datasets. The size of the output buffer can be changed in `DSNZPARM OUTBUFF`.
- Increase the active log size. If logs fill up faster than they can be archived, performance suffers.
- Determine the optimal number of DB2 active logs based on peak EIM loads plus a cushion.

- Place the active log data sets on separate dedicated volumes.
- Place archive logs on a virtual tape system (VTS), or disk if VTS is not available.
- Define all table spaces (base tables and interface tables) with compression, which reduces logging, and decreases I/O.
- Consider data sharing when a single subsystem cannot handle the logging tasks.

## Using the RUNSTATS Utility

If EIM performance degrades over time, consider running RUNSTATS against the base tables to update the optimizer statistics.

If the parameter `UPDATE STATISTICS` is set to `TRUE` in your `.ifb` file, the EIM process invokes the DSNUTILS stored procedure. This allows EIM to perform RUNSTATS dynamically on temporary columns to get optimal statistics numbers in the DB2 catalog. The IBM DSNUTILS stored procedure must be installed and the OS/390 WLM manager should be enabled.

To execute EIM with `UPDATE STATISTICS` set to `TRUE`, the user account must have EXECUTE privileges on the DSNUTILS stored procedure and SYSADM, STATS, or DBADM privileges on the tables accessed.

**NOTE:** Set `UPDATE STATISTICS = TRUE` only once to collect proper statistics for each given EIM process. As soon as this is done, reset `UPDATE STATISTICS` to `FALSE` for that process.

For more information on RUNSTATS, see [Chapter 11, “Maintenance Considerations for zSeries.”](#)

## Reviewing the SQLPROFILE Log

The `SQLPROFILE` parameter in the header of your EIM configuration (`.ifb`) file designates the file to which EIM writes an analysis of the worst-performing SQL queries overall (by total time) and the worst-performing statements in each step of the EIM process (also by total time), for example:

```
[Siebel Interface Manager]
```

```
PROCESS = Import Products
```

```
SQLPROFILE = topsql.log
```

Review this log file after the test run of each EIM process to determine where potential bottlenecks exist in your process. Based on the results of this log, either adjust mapping and IFB parameters, or perform normal DB2 tuning (such as adding an index to improve the access path).

## Improving EIM Performance During the Import Process

The following procedures and techniques can improve EIM performance during import processes, particularly during your initial load of data into Siebel eBusiness Applications.

You should follow the recommended import order as described in the *Siebel Enterprise Integration Manager Administration Guide* for both initial and ongoing EIM loads.

## Using the Default Column to Improve Throughput

If certain values in your database will be constant across all rows of a table, use the `DEFAULT COLUMN` parameter to fill these rather than loading them through the EIM tables to improve throughput.

## Tuning Your EIM Import Process

Review and tune your EIM process for each set of business data you load, for example, Opportunities, Contacts, or Products.

### *To tune each EIM import process*

- 1 Load a single batch with the following flags set in your configuration file:

- Error flag = 1
- SQL flag = 8
- Trace flag = 3

This produces a log file with SQL statements and information about how long each statement took to execute.

- 2 Identify SQL statements that might be taking too long to execute.
- 3 Using the DB2 EXPLAIN utility, populate the explain table with information on how the DB2 Optimizer is executing each of these SQL statements. Pay particular attention to any changes in the default optimization level in the EIM log.
- 4 Based on the Access Plan and other information from the EXPLAIN output, determine required indexes for EIM and base tables.
- 5 Run EIM with parameters to reach each partition and repeat single process tuning.
- 6 Perform .i fb file tuning for each process. Remove unnecessary foreign keys.
- 7 Execute the REORG utility on base tables to build a compression dictionary.
- 8 Perform parallel process tuning.
- 9 Perform buffer pool tuning.
- 10 Execute STOSPACE on base tables. This collects storage information on DB2 objects.
- 11 Perform RUNSTATS on base and EIM tables.
- 12 Perform EXPLAINS to verify access path (required indexes).
- 13 Modify statistics if required.
- 14 Drop unused indexes on base and EIM tables.

- 15 Check space and resize table spaces if necessary.

## Initial Load Considerations

There are certain considerations that are only applicable during an initial load of data into your Siebel application.

### Drop Unused, Non-Unique Indexes

The initial load is typically a database-intensive process. Each row that is inserted into the base table requires modifications not only to the table itself, but to all its affected indexes. However, most of these indexes are never used during an initial load process.

Performance can usually be improved by determining which base table indexes are unused by EIM during the import process and dropping them prior to the initial load. These indexes must later be recreated by your DB2 database administrator (DBA).

### Dropping Unused Triggers

Dropping unused triggers can improve import performance by reducing the system overhead used to verify trigger constraints. You can use the Generate Triggers (GenTrig) component from Siebel Server Tasks to drop the triggers and to re-create them later on. This component has to be in the “Enabled” state. For details on how to work with Generate Triggers, see the *Siebel Business Process Designer Administration Guide*.

**NOTE:** If you are employing partitioning, do not drop triggers that are used for partitioning purposes. If you do your EIM process fails, or it inserts all the data in a single partition.

### Optimizing Free Space Parameters

If you will use your Siebel application primarily for queries, updates and deletions, you should alter your table spaces and indexes to provide optimal insert and update performance.

To improve the EIM import and update SQL performance, set PCTFREE for table spaces and indexes to a value of 20 or 30 prior to the EIM initial load, and maintain these settings at 20 or 30 for subsequent loads.

#### ***To improve the insert performance on base tables during EIM initial load***

- 1 Set PCTFREE to zero for data and sequential index.
- 2 Set FREEPAGE to zero for data and index.
- 3 Use Page Lock if you are sequentially inserting many rows or pages.
- 4 For a data sharing environment, set the member cluster option to reduce spacemap and data page P-Lock contention.

You should alter base table spaces to PCTFREE and FREEPAGE values of 20 or 30 for subsequent import processes. You should also load or reorganize any altered items before the new values take effect.

The following example demonstrates how to set these values for your table spaces and indexes. Substitute your own table space and index names for your implementation.

```
ALTER TABLESPACE S1DB0401.H0401000 PART 1 PCTFREE 20;
ALTER TABLESPACE S1DB0401.H0401000 PART 2 PCTFREE 20;
```

```
ALTER INDEX S1DB0401.S_ADDR_ORG_M6 PCTFREE 20;
ALTER INDEX S1DB0401.S_ADDR_ORG_P1 PCTFREE 20;
ALTER INDEX S1DB0401.S_ADDR_ORG_P99 PART 1 PCTFREE 20;
ALTER INDEX S1DB0401.S_ADDR_ORG_P99 PART 2 PCTFREE 20;
```

## Disabling Transaction Logging

Siebel transaction logging is unnecessary during an initial EIM load, and should be disabled by setting LOG TRANSACTIONS = FALSE in your .i fb file. For more information on transaction logging in EIM, see the *Siebel Enterprise Integration Manager Administration Guide*.

# Considerations for EIM Tables

Consider the following issues when working with EIM tables.

## Updating the S\_LST\_OF\_VAL Table

The BU\_ID column in the S\_LST\_OF\_VAL base table can have only one or very few distinct values. When this happens, the DB2 optimizer often needlessly performs a table scan through all rows in the S\_LST\_OF\_VAL table.

To avoid this problem and speed up the query, modify the statistics data by running the following SQL statements:

```
update sysibm.sysindexes set firstkeycardf=1000 where
name=' S_LST_OF_VAL_M2' ;

update sysibm.syscolumns set colcardf = 1000 where
tbname=' S_LST_OF_VAL' and name=' BU_ID' ;
```

**NOTE:** Depending on the data you are working with, you might need to run other SQL statements ahead of time.

## Resolving Performance Degradation During the Load Process

When executing the EIM load process, performance might degrade noticeably over time, after subsequent batch loads. This performance degradation is often due to outdated statistics on the tables being loaded. To resolve the problem, update statistics on the target base tables.

## Resolving Errors in the EIM Process

Perform the steps below to resolve errors that might occur during your EIM process:

- If EIM aborts with the following generic error, then the user account running EIM does not have EXECUTE privileges on the DSNUTILS stored procedure:

```
DSNU060I database_alias DSNUGMAP - USER username NOT AUTHORIZED FOR RUNSTATS UTILITY
ON
```

See [“Using the RUNSTATS Utility” on page 180](#) for information about required permissions. Alternatively, you can run EIM with the following setting:

```
UPDATE STATISTICS = FALSE
```

- If your EIM process fails with the following error message, then the DSNZPARM parameter RETVLCFK might be set incorrectly.

```
EIM-00205 Failed To Load the Application Dictionary
```

Siebel eBusiness Applications require the setting RETVLCFK=NO. For information on DSNZPARM settings see [Chapter 4, “Preparing for Implementation on the DB2 Host.”](#)

- If EIM terminates during a DELETE EXACT process involving deletion of child records, the following error indicates that the maximum allowable locks were exceeded:

```
EIM Fails - Max Locks Exceeded on DELETE EXACT
```

This causes the child records to be orphaned, because the delete to the parent table has already been committed.

You can avoid this error by specifying the following in your .i fb file:

```
COMMIT EACH PASS = FALSE
```

```
COMMIT EACH TABLE = FALSE
```

```
ROLLBACK ON ERROR = TRUE
```



# A Deployment Planning Worksheet

Each time you install a new Siebel eBusiness application using IBM DB2 UDB for z/OS and OS/390, you must make copies of this worksheet and give it to each member of the deployment team. This worksheet includes the following sections:

## Master Worksheet

- “Team Lead Summary”
- “DB2 Connect Information”
- “Siebel Database Server Installation Information”

### Team Lead Summary

|                                              |  |
|----------------------------------------------|--|
| <b>Deployment Team Lead:</b>                 |  |
| <b>Siebel Administrator:</b>                 |  |
| <b>Privileged User/Siebel Database User:</b> |  |
| <b>DB2 Systems Programmer (SYSADM):</b>      |  |
| <b>DB2 Database Administrator (DBADM):</b>   |  |
| <b>Security Administrator:</b>               |  |
| <b>z/OS and OS/390 System Programmer:</b>    |  |
| <b>Midtier System Administrator:</b>         |  |

## DB2 Connect Information

DB2 Host Name/IP Address:

■ DB2 Port Number:

## Siebel Database Server Installation Information

Siebel Gateway Name Server  
Name:

Enterprise Server Name:

Siebel Server Name:

Path to Siebel File System:

Database Alias:

Schema Qualifier ID  
(Max. 8 characters)

Security Group ID /  
Grantee

Siebel Administrator User  
Name

Siebel Administrator  
Password

Siebel Administrator User  
Group

Siebel User Group ID  
(Max. 8 characters)

EIM User Group ID  
(Max. 8 characters)

Database Name Prefix  
(Max. 6 characters)

Name of Storage Control  
File Used

4-KB Buffer Pool

16-KB Buffer Pool

32-KB Buffer Pool

Index Buffer Pool

Code Page/CCSID

**NOTE:** The Security Group ID is also known as the secondary authorization ID.

# Index

## A

**ASCII-based CCSID, about choosing for implementation** 35

### **Assignment Manager**

customize applications, using for 162

table partitioning strategies 64

**authorizations, based on implementation task (table)** 21, 23

**auxiliary objects, naming conventions** 51

## B

**buffer pools, allocating space and storage groups** 40

## C

**CCSID, about choosing for implementation** 35

**Character Large Object support, about .ctl file** 145

See OS/390 development database, cloning  
or z/OS development database,  
cloning

## D

### **data**

See Siebel EIM, using to migrate data for DB2

### **data mapping**

MS\_IDENT, caution about loading 178

necessary data, about loading 178

presorting data during load, about 178

### **database**

schema changes, about updating local  
databases 162

schema extensions, applying to development  
database 148

target database, applying schema changes  
to 157

target database, migrating rules 161

target database, preparing 150

### **Database Manager Configuration Parameters**

See DSNZPARM parameters

### **database objects**

distribution, about planing for 40

estimating number required, about and  
table 41

Siebel tables. frequently used and largest

(table) 41

### **Database Server software**

UNIX, installing under 77

Windows, choosing different directory 74

Windows, installing under 73

### **Database Storage Configuration**

See Siebel Database Storage Configurator

### **DB2 catalog**

storage file, extracting under UNIX 104

storage file, extracting under Windows 102

### **DB2 connect**

Connect Personal Edition (PE), used to  
support Siebel Dedicated Web  
Client 29

editions of 25

EE, using to support Siebel Web Client 28

setup instructions for configuration,  
about 28

setup options 26

### **DB2 global statement caching**

about using 174

cached statements, invalidating 174

### **DB2 host, preparing for implementation on**

buffer pools, allocating space and storage  
groups 40

database CCSID, choosing 35

database object, estimating number  
required 41

database objects, about planning distribution  
of 40

database objects, frequently used and largest  
Siebel tables (table) 41

database, about connecting to 26

DB2 connect, setup options 26

DB2, setup instructions 28

odbcsql.exe, using to verify network  
connectivity 173

storage space, estimating 39

system connectivity architecture 25

### **DB2 operating system, about security** 16

### **DB2 UDB, file installation**

Siebel Database Server software, purpose of  
installing 72

Siebel Database server, preinstallation  
tasks 72

Siebel Schema, preparing for installing 72

UNIX installation directory structure 81

- UNIX, installing Database Server software under 77
  - Windows installation directory structure 80
  - Windows, installing Database Server software under 73
  - DB2 UDB, from development to production**
    - See also OS/390 development database, cloning; z/OS development database, cloning; Siebel Tools
    - Siebel Schema, applying schema changes to target db 155
    - Siebel Schema, applying schema extensions to development database 148
    - Siebel Schema, making custom extensions to 146
    - Siebel Schema, migrating customizations to target database 151
    - Siebel Schema, preparing target database 150
    - Siebel Schema, tasks overview 147
  - DB2 UDB, post-installation tasks**
    - database, validating 130
    - loadstat.sql script, editing and running 129
    - Siebel File System, populating 137
    - table privileges, granting 129
    - UNIX, importing new language 136
    - UNIX, installing multilingual seed data 133
    - Windows, importing new language 134
    - Windows, installing multilingual seed data 132
  - DB2 UDB, Siebel Repository**
    - installation and configuration tasks, table of 121
    - log files, reviewing 128
    - log files, troubleshooting 129
    - Windows, importing under 122
  - DB2 UDB, Siebel Schema installation**
    - DB2 catalog, extracting storage file under UNIX 104
    - DB2 catalog, extracting storage file under Windows 102
    - DDL, applying on DB2 host 95
    - installation and configuration tasks, table of 83
    - installation process, about 84
    - rerunning the installation 120
    - Siebel Database Storage Configurator, using to automate configuration 53, 108
    - Siebel tables, validating 99
    - storage control file, about preparing and methods 101
    - UNIX, custom installation 118
    - UNIX, reviewing log files 119
    - Windows, custom installation 115
    - Windows, reviewing log files 117
    - Windows, standard install option 84
  - DB2/Connect installation** 26
  - DBADM/CREATEDBA, about** 19
  - dbchck, running to validate database** 130
  - dbconf.xls**
    - See Siebel Database Storage Configurator
  - Deployment Planning Worksheet**
    - about using 185
    - Master Worksheet 185
  - development database, applying schema extension to** 148
  - DSNZPARM parameters**
    - cached statements, invalidating 174
    - DB2 global statement caching, about 174
    - NUMLKUS considerations 38
    - Siebel dynamic SQL, about 170
- E**
- EBCDIC-based CCSID, about choosing for implementation** 35
  - EIM tables partitioning strategies** 64
  - external security adapter**
    - OS/390, using 17
    - z/OS, using 17
- G**
- Generate DDL Into Files, migrating customization to target db** 151
  - global statement caching**
    - See DB2 global statement caching
  - grantstat.sql script, running** 129
  - group ID protection scheme, about** 17
- I**
- importing, improving EIM performance**
    - DEFAULT COLUMN, using to improve throughput 180
    - EIM import process, tuning 181
    - transaction logging, disabling 183
    - unused non-unique indexes, dropping 182
    - unused triggers, dropping 182
  - indexes, reorganizing fragmented indexes** 164
  - installation**
    - See also security
    - DBADM/CREATEDBA, roles and permissions required for 19
    - group ID protection scheme, about 17
    - SYSADM, roles and permissions required for 18

**L****local environment, updating** 148**log files**

- Siebel Repository, reviewing 128
- Siebel Repository, troubleshooting 129
- UNIX, reviewing 119
- Windows, reviewing 117

**M****maintenance**

- fragmented tables and indexes, reorganizing 164
- statistics, updating 163

**migrating rules** 161**Mobile Web Clients**

- See Siebel Mobile Web Client

**N****NUMLKUS, considerations when setting up DSNZPARM parameters** 38**O****odbcsql.exe, using to verify network connectivity** 173**OS/390 development database, cloning**

- See *also* individual DB2 UDB entries
- about 158
- customizing using Assignment Manager and Siebel Workflow 162
- migrating rules 161
- schema changes, about applying to other databases 162
- storage control file, extracting from development db 158
- UNIX, generating DDL file 160
- Windows, generating DDL file 157, 158

**OS/390 security**

- See *also* individual DB2 UDB entries
- external security adapter, using 17
- passwords, about changing 16

**P****password, changing and expiration** 16**performance**

- fragmented tables and indexes, reorganizing 164
- statistics, updating 163

**R****Run DDL Automatically, migrating customizations to target db** 151**RUNSTATS**

- migrate data, using for 180

**S****S\_ESCL\_LOG table**

- See Assignment Manager table

**S\_ESCL\_REQ table**

- See Assignment Manager table

**SADMIN, about changing security** 23

- See *also* installation
- data transmission, about 18
- DB2 operating system 16
- password change and expiration, about 16
- required authorizations, table of 21, 23
- Siebel application data security, about 15

**seed data**

- UNIX, installing multilingual seed data 133
- Windows, installing multilingual seed data 132

**Siebel application data security, about Siebel Database Server** 15

- UNIX, installation directory structure 81
- Windows, installation directory structure 80

**Siebel Database Server for DB2, installing** directory, choosing different 80**Siebel Database Storage Configurator**

- about 53, 108
- CLOBs, activating 113
- control files, configuring 109
- partitions, decreasing the number of 111
- partitions, increasing the number of 112
- worksheet, validating 111

**Siebel Dedicated Web Client**

- DB2 Connect PE, using to support 29

**Siebel dynamic SQL, about** 170**Siebel EIM, using to migrate data for DB2**

- application dictionary, failed to load message 184
- data mapping, planning 178
- database logging, optimizing 179
- EIM aborts with not authorized to for RUNSTATS 184
- EIM batch size, choosing optimal size 179
- import considerations 180
- load process, performance degradation during 184
- maximum locks exceeded, about 184
- parallel streams, choosing optimal number 179
- partitioning scheme, choosing 177
- RUNSTATS, using 180
- S\_LST\_OF\_VAL Table, updating 183
- setting up for 177

SQLPROFILE log, reviewing 180

**Siebel Enterprise Integration Manager**  
See Siebel EIM, using to migrate data for DB2

**Siebel File System, populating** 137

**Siebel Mobile Web Client**  
data transmission security, about 18

**Siebel objects naming conventions** 50

**Siebel Remote Transaction Logging table**  
performance considerations 63  
transaction logging, turning off 63

**Siebel Repository**  
See DB2 UDB, Siebel Repository

**Siebel Schema**  
changes, about applying to local  
databases 162  
extensions, applying schema changes to  
target db 155  
extensions, applying to development  
database 148  
extensions, making custom extensions  
to 146  
extensions, migrating customization to target  
database 151  
extensions, preparing target database 150  
extensions, tasks overview 147  
migrating rules 161

**Siebel storage control file templates**  
about 44, 45  
buffer pools 47  
object naming conventions 50  
partitioning, changing partition keys 62  
partitioning, deciding what to and  
example 58  
prepartitioned Siebel tables, table of 65  
storage groups, about 47  
table spaces, about 49  
tables, without page-level locking (table  
of) 49

**Siebel templates**  
partitioning, changing partition keys 62  
partitioning, deciding what to and  
example 58  
prepartitioned Siebel tables, table of 65

**Siebel Tools**  
See *also* Assignment Manager  
Character Large Object (CLOB) support,  
about 145  
configuration file parameters, table of 140  
database options, setting 140  
index truncation, about 141  
table grouping file names, about 141  
unused indexes, about deleting 141  
VARCHAR field length, about reducing 141

**Siebel Web Client**

data transmission security, about 18  
DB2 connect EE, using to support 28

**Siebel Workflow, using to customize  
applications** 162

SQLPROFILE log, reviewing 180

SSEROLE, about changing names 23

statistics, updating 163

storage control file, about preparing and  
methods 101

storage space, estimating 39

synchronizing, about data transmission  
security 18

SYSADM, about 18

## T

### tables

See *also* Siebel EIM, using to migrate data for  
DB2

fragmented tables, reorganizing 164

### tables, partitioning

about 55

Assignment Manager and Workflow tables,  
about and partitioning strategies 64

EIM tables, partitioning strategies 64

partitioning index, about and diagram  
example 56

Siebel Remote Transaction Logging table,  
about and strategies 63

triggering, about using for performance 62

### target database

customization, migrating from development  
to target db 151

migrating rules 161

preparing target db 150

schema changes, applying to target db 155

### templates

See Siebel storage control file templates

### Time Sharing Option (TSO), about changing passwords

### transaction logging, disabling to improve EIM performance

### triggers

unused triggers, dropping to improve EIM  
performance 182

### tuning

fragmented tables and indexes,  
reorganizing 164  
statistics, updating 163

## U

### Unicode conversion tables

### UNIX

Database Server software, installing

- under 77
- DB2 UDB, Siebel Schema installation, custom installation 118
- DB2 UDB, Siebel Schema installation, reviewing log files 119
- DDL, generating file 160
- log files, acceptable errors 119
- multilingual seed data, installing 133
- schema, migrating 154
- Siebel Database Server, installation directory structure 81
- Siebel Repository, importing new language 136
- Siebel Repository, reviewing log files 128
- Siebel Repository, troubleshooting log files 129
- storage file, extracting from the DB2 catalog 104

## W

### Web Client

See Siebel Web Client

### Windows

- Database Server software, choosing different directory 74
- Database Server software, installing under 73
- DB2 UDB, Siebel Schema installation, custom installation 115
- DB2 UDB, Siebel Schema installation, reviewing log files 117

- DDL file, generating 157, 158
- multilingual seed data, installing 132
- schema, migrating 151
- Siebel Database Server, installation directory structure 80
- Siebel Repository, importing 122
- Siebel Repository, importing new language 134
- Siebel Repository, reviewing log files 128
- Siebel Repository, troubleshooting log files 129
- Siebel Schema on DB2, standard install option 84
- storage file, extracting from the DB2 catalog 102

**Workflow table partitioning strategies worksheet, Deployment Planning** 64 185

## Z

### z/OS development database, cloning

See *also* individual DB2 UDB entries about 158

storage control file, extracting from development, db 158

UNIX, generating DDL file 160

Windows, generating DDL file 157, 158

### z/OS security

See *also* individual DB2 UDB entries external security adapter, using 17

**zSeries customers, about changing passwords** 16

