Implementing Siebel Business Applications on DB2 UDB for z/OS

Version 7.8, Rev. B

August 2005

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

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

Proprietary Information

Siebel Systems, Inc. considers information included in this documentation and in Siebel 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 7: Installing the Siebel Database Server Files at Midtier

Chapter 8: Installing the Siebel Schema on the DB2 Host

Chapter 9: Siebel Repository Import and Postinstallation Tasks

Chapter 10: Customizing a Development Environment

Chapter 11: Maintenance Considerations for z/OS

Chapter 12: Migrating Data Using Siebel EIM

Appendix A: Deployment Planning Worksheet

Index

1 What's New in This Release

What's New in Implementing Siebel Business Applications on DB2 UDB for z/OS, Version 7.8, Rev. B

Table 1 lists changes described in this version of the documentation to support Release 7.8 of the software.

Table 1. What's New in Implementing Siebel Business Applications on DB2 UDB for z/OS, Version 7.8, Rev. B

Topic	Description
"Process of Extending the Siebel Schema" on page 141	The reference to the Siebel Tools project name, Newtable, was changed to Table Common.
"DB2 Statistics for Siebel Business Applications" on page 157	This section was changed to state that if you are experiencing slow queries, the DSTATS utility should be used to collect column distribution statistics.

What's New in Implementing Siebel Business Applications on DB2 UDB for z/OS, Version 7.8, Rev. A

Table 2 lists changes described in this version of the documentation to support Release 7.8 of the software.

Table 2. What's New in Implementing Siebel Business Applications on DB2 UDB for z/OS, Version 7.8, Rev. A

Topic	Description
"The Data Distribution Facility and Workload Manager" on page 33	This section was amended to clarify that Workload Manager must be installed and configured if the Siebel application stored procedures and user defined functions are to work.
"Storage Control File Names" on page 135	The storage control file was previously referred to as the table groupings file. This section title was changed from "Specifying Table Grouping File Names" to reflect current terminology.
"Acceptable Errors and the errors.rtf File" on page 116.	This section was moved from Chapter 9, "Siebel Repository Import and Postinstallation Tasks" to Chapter 8, "Installing the Siebel Schema on the DB2 Host" because this information is relevant when installing the Siebel Schema.

Additional Changes

This version of the documentation also contains the following general changes:

- Grammar, punctuation, and general rewriting
- Minor structural changes within chapters to reduce redundancy and improve readability.

What's New in Implementing Siebel Business Applications on DB2 UDB for z/OS, Version 7.8

Table 3 lists changes described in this version of the documentation to support Release 7.8 of the software.

Table 3. New Features in Implementing Siebel Business Applications on DB2 UDB for z/OS, Version 7.8

Topic	Description
"Configuring DB2 Connect EE, ASE or UE to Support Siebel Developer Web Client" on page 27	Siebel Dedicated Web Client has been renamed Siebel Developer Web Client. This client type is no longer supported for deployment to end users but is supported for administration and development use.
"About Unicode Conversion Tables" on page 32	Information was added on installing Unicode Conversion Services. These services are required if Siebel applications are to communicate with a DB2 database that does not use Unicode character set.
"DDSNZPARM Parameter Settings for Siebel Applications" on page 34	New information has been added to the guide to reflect that DB2 for z/OS v8 is supported in this release. DB2 for z/OS and OS/390 v7 is still supported; information that is specific to a particular version of DB2 for z/OS is noted.

2 Preparing to Deploy Siebel on DB2 for z/OS

There are a number of issues you need to consider when preparing your implementation of Siebel Business Applications using IBM DB2 UDB for z/OS (hereafter referred to as DB2 for z/OS). These issues are covered in depth throughout this guide.

This guide also describes how to install the Siebel Database Server and the Siebel Schema. For information on installing the core Siebel products, see the *Siebel Installation Guide* for the operating system you are using.

This chapter contains the following topic:

"Process of Preparing for a Siebel Application Deployment on DB2 for z/OS"

Process of Preparing for a Siebel Application Deployment on DB2 for z/OS

Preparing to deploy a Siebel application on DB2 for z/OS involves the following steps:

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 DB2 databases, the requirements include:
   - Applying required DB2 maintenance as listed in the IBM informational APAR ii13487.
   - Installing a supported version of DB2 Connect and the appropriate IBM Fix Pack as specified in the *System Requirements and Supported Platforms* on Siebel SupportWeb.
   - Installing Unicode Conversion Services.
2. Review the Siebel architecture of the DB2 for z/OS database server.
3. Decide on the DB2-related architecture and other related choices:
   - Understand and design the database storage layout for the Siebel Schema.
   - Review possible table space partitioning.

     If you plan on using batch processing through EIM, ensure the partitioning scheme is appropriate for both batch and online activities.
   - Decide whether to use a separate or shared DB2 subsystem for production.
   - Configure the DB2 Data Distribution Facility and the z/OS Workload Manager (WLM).
   - Review and set up the required and recommended DB2 system parameters (DSNZPARMs).
   - Decide whether to select an ASCII or EBCDIC encoding scheme code page.
   - Decide whether to implement Coordinated Universal Time (UTC).

   - Decide which security scheme to use (native DB2 security or Lightweight Data Access Protocol (LDAP)).
   - Decide on backup and recovery mechanisms.
   - Decide on maintenance procedures, such as RUNSTATS and REORG.
   - Ensure you know how to deploy the user-defined functions (UDFs) and stored procedures that Siebel applications require.
   - Specify the appropriate Siebel parameters and environment for your deployment after the installation is completed.
   - 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 database administrators (DBAs) to support the Siebel application deployment.
   - Establish good working relationships with midtier 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 that in the Siebel application:
     - There are many objects, some of them unused.
     - The Siebel data model uses different data types, mainly VARCHAR.
     - Dynamic SQL is generated versus traditional SQL precisely coded by developers.

3 Security Concepts for z/OS

This chapter discusses the security concepts that are important in running Siebel Business Applications on DB2 for z/OS. This chapter also describes the roles and permissions you require 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 the Database Server.

For a general discussion of Siebel architecture, 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 describes the following topics:

- "About Siebel Application Data Security" on page 13
- "Operating System Security" on page 14
- "Password Change and Expiration" on page 14
- "Changing Passwords from Remote Clients" on page 14
- "z/OS Security" on page 15
- "Data Transmission Security for Siebel Clients" on page 16
- "Required Authorizations" on page 19

About Siebel Application Data Security

All users must provide a user name and password to connect to Siebel Business Applications, regardless of whether they access them through the Siebel Developer 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 access to data by association with Positions and Responsibilities.

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

Siebel Business 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 the *Applications Administration Guide* and the *Security Guide for Siebel Business Applications*.

Operating System Security

The DB2 Connect middleware passes the user name and password used to access Siebel Business Applications to DB2 to establish a connection. The user name and password can be user credentials supplied by the external security adapter, if you are using an external security adapter.

The Security Administrator (or someone who has been delegated this authority) must preregister all user names and passwords passed through DB2 Connect with the z/OS security package in use, for example, RACF, ACF/2, or TOP SECRET. The z/OS Security Administrator can associate the user name with a secondary authorization ID (also known as the Security Group ID on the "Deployment Planning Worksheet") within the z/OS security package to simplify database security privilege administration.

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

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

Password Change and Expiration

Because of security constraints, if you are using Siebel applications on z/OS, user passwords might be setup to periodically expire.

- **When a password has expired.** In Siebel Business Applications, you can change expired passwords from the login window without administrative intervention.
- **When a password has not yet expired.** If a password has not yet expired, you can change it through the Change Password client function.

Changing Passwords from Remote Clients

To change z/OS 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 DSNZPARM EXTSEC=YES.

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

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

z/OS 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 Web site. 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 have a primary authorization ID.
- **Secondary authorization ID.** Generally known as a group ID. Siebel applications effectively use secondary authorization IDs that are 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

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

NOTE: When you install the Siebel Schema, you are prompted to enter a Security Group ID/Grantee. This is the same as a secondary authorization ID.

During the Siebel Schema installation process, you can 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 authority 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 is among the list of secondary authorization IDs for all users of the applications.

Grant statements for additional secondary authorization IDs.

You must create secondary authorization IDs separately. Siebel Business Applications include the grantstat.sql script; this script generates grant statements which allow access to interface tables. For a discussion of the grantstat.sql script, see "Granting Table Privileges" on page 125.

Either the table owner, or users with DBADM or SYSADM privileges, must execute the grant statements. 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 repository, a protocol for storing and retrieving directory-related information that includes authentication services. LDAP can reside on the mainframe.

Administration is easier if you use an external security adapter 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 on implementing an external security adapter, refer to the *Security Guide for Siebel Business Applications*.

Data Transmission Security for Siebel Clients

Siebel Web Clients and Mobile Web Clients access Siebel Business Applications over the Internet and Mobile Web Clients can also 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 for z/OS 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 Business Applications on a DB2 database:

- SYSADM
- DBADM
- CREATEDBA

SYSADM Privileges Used for Connecting to DB2

A DB2 subsystem is a prerequisite for installing Siebel Business Applications. Although you do not need to use an ID with SYSADM privileges to install Siebel Business Applications, you might need such an ID to create underlying DB2 resources. For detailed information on setting up a DB2 subsystem for Siebel Business Applications, see Chapter 4, “Preparing for Implementation on the DB2 Host.”

Functions that require SYSADM authority and that are necessary when you install Siebel Business Applications on DB2 for z/OS include:

- Allocating and accessing buffer pools
- Allocating and accessing storage groups
- Granting CREATEDBA or DBADM authority to the Siebel user ID used for the Siebel database installation
- Creating user-defined functions and stored procedures.

DBADM/CREATEDBA Privileges Used for Connecting to DB2

To install the Database Server 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 Business 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.SYSTRIGGERS

To grant SELECT authority to access the DB2 Catalog

- Use this 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

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 located in the `dbsrvr/db2390` directory, which is created during the database server installation. For information on using this script, see "Loading Statistics" on page 125.

NOTE: Updating DB2 catalog statistics is optional.

Granting Authorization to Views in DB2 for z/OS v8

GRANT VIEW statements can fail on DB2 for z/OS v8 if you use an external security manager, such as RACF, to protect DB2 resources, and internal security authorizations are not also in place. Such failure may occur because—when a GRANT VIEW statement is issued on DB2 for z/OS v8—DB2 only carries out internal database security checks before giving authorization to the view. If an internal security mechanism defining privileges to views does not exist, because these privileges are defined using an external security manager, the GRANT VIEW statement fails.

This process does not occur on DB2 for z/OS v7 and is not carried out when GRANT statements for tables are issued.

Siebel Systems issues GRANT VIEW statements only in the ddlview.sql file; this contains the DDL to create the Siebel Schema.

Required Authorizations

Table 4 lists the authorizations that are necessary in implementing Siebel Business Applications on DB2 for z/OS.

Table 4. DB2 Authorizations Required to Implement Siebel Business Applications

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. DB2 Authorizations Required to Implement Siebel Business Applications

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;`

Table 4. DB2 Authorizations Required to Implement Siebel Business Applications

Task	Authorization Required	Task Command Example
Grant SELECT on catalog tables.	SYSADM, SYSCTRL	`GRANT SELECT ON SYSIBM.SYSTABLES TO SIEBTO;`

Create User-Defined Functions	SYSADMIN, DBADM	`CREATE FUNCTION SIEBEL.NEXTSEQ (INTEGER)`
		`RETURNS  SYSIBM.INTEGER`
		`SPECIFIC SIEBEL.NEXTSEQ`
		`EXTERNAL NAME 'NEXTSEQ'`
		`LANGUAGE C`
		`PARAMETER STYLE DB2SQL`
		`NOT DETERMINISTIC`
		`FENCED RETURNS NULL ON NULL INPUT`
		`NO SQL`
		`NO EXTERNAL ACTION`
		`SCRATCHPAD 100`
		`NO FINAL CALL`
		`DISALLOW PARALLEL`
		`NO DBINFO`
		`WLM ENVIRONMENT DB27WLM`
		`STAY RESIDENT NO`
		`PROGRAM TYPE SUB`
		`SECURITY DB2`
		`INHERIT SPECIAL REGISTERS`

Table 5 lists the authorizations that users of Siebel Business Applications might need. Your enterprise may have unique role names that it assigns with the authorities listed in this table. Therefore, the role names in Table 5 are 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	Owns the schema objects (created by DBADM) that are 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
			`RUNSTATS TABLESPACE D0010677.H0677000 TABLE(SIEBEL.S_CONTACT)FREQVAL COUNT 10`
			`CREATE TRIGGER SIEBEL.PTH0477`
			`NO CASCADE BEFORE INSERT ON`
			`SIEBEL.S_ASSET`
			`REFERENCING NEW AS N`
			`FOR EACH ROW MODE DB2SQL`
			`WHEN (`
			`N.ROW_ID IS NOT NULL)`
			`BEGIN ATOMIC`
			`SET N.PARTITION_COLUMN = RIGHT`
			`(N.ROW_ID, 2);`
			`END`

NOTE: SSEROLE and SADMIN are Siebel defaults for the Siebel user group and the Siebel administrator. However, you can change these names to meet your organization’s naming conventions.

4 Preparing for Implementation on the DB2 Host

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

This chapter consists of the following topics:

System Connectivity Architecture

Siebel Business Applications communicate with DB2 for z/OS through IBM DB2 Connect middleware.

NOTE: IBM DB2 Connect v7.2 is not supported in Siebel 7.8. Verify that you are using IBM DB2 Connect v8.2.

The four 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. This edition is installed on a midtier server, such as a Siebel Server machine.
- **DB2 Connect Application Server Edition (ASE).** Provides the functionality of DB2 Connect Enterprise Edition but is licensed and priced differently.
- **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 Developer Web Client on a workstation. This edition is installed on a user's local workstation.

DB2 Connect EE, ASE, or UE can reside either on the same machine as the 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, ASE or UE broker the connections to DB2 for multiple database clients. See Table 6 on page 24 for DB2 Connect configurations.

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

Siebel Servers and Siebel Developer 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.) For instructions on installing TCP/IP, refer to the vendor documentation on the IBM Web site.

Connecting to the Database Using DB2 Connect

You install the Siebel Database Schema on DB2 for z/OS using the Database Server configuration wizard on a client Siebel Server. See Chapter 8, “Installing the Siebel Schema on the DB2 Host” for a description of the installation process. After installation, you execute SQL on DB2.

Configuration Options for DB2 Connect

When using DB2 Connect, your configuration options depend on whether you are deploying Siebel Business Applications on a Web Client or on a Developer Web Client. This is shown in Table 6.

Table 6. DB2 Connect Configurations

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

Figure 1 illustrates some of the configurations possible with DB2 Connect Personal Edition (PE) and DB2 Connect Enterprise Edition (EE). DB2 Connect PE runs on the workstation only; DB2 Connect EE can run on the same machine as the Siebel Server or on a different machine. The configuration you choose largely depends on the types of Siebel clients your enterprise supports.

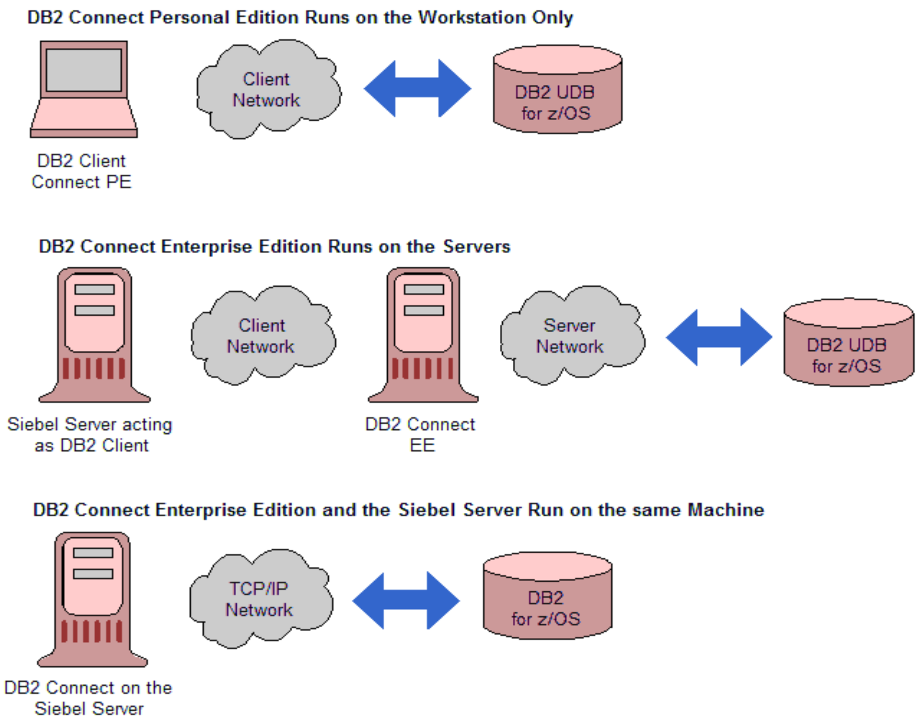

DB2 Connect Personal Edition Runs on the Workstation Only
Client Network
DB2 UDB for z/OS
DB2 Client Connect PE
DB2 Connect Enterprise Edition Runs on the Servers
Client Network
Server Network
DB2 UDB for z/OS
Siebel Server acting as DB2 Client
DB2 Connect EE
DB2 Connect Enterprise Edition and the Siebel Server Run on the same Machine
TCP/IP Network
DB2 for z/OS
DB2 Connect on the Siebel Server

Figure 1. DB2 Connect Enterprise Edition and Personal Edition Configurations

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

Installing Required IBM Fix Packs

After you have installed DB2 Connect, install any relevant, certified IBM Fix Pack. Not all Fix Packs are certified to run with Siebel applications. To find out which Fix Packs are certified, refer to *System Requirements and Supported Platforms* on Siebel SupportWeb, the vendor documentation, or the following IBM URL:

```
ftp://ftp.software.ibm.com/ps/products/db2/fixes/english-us/siebel/siebel7/
Siebel7CertifiedFixpaks.txt.
```

Refer to IBM documentation for installing the Fix Pack for your environment.

NOTE: When using the UNIX operating system, if you install a Fix Pack after you create the DB2 database instance, you must update the instance using the db2iupdt command.

You can verify the currently installed version of DB2 Connect and the Fix Pack by running the command db2level from the DB2 Command Window. In a UNIX environment, this command is run from a UNIX shell after sourcing the appropriate db2profile. Make a note of the information and compare the Informational tokens from the output against *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 in this section that apply to your chosen DB2 Connect configuration.

- **Define a database alias.** Define a database alias with the appropriate 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 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 for z/OS.

  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 sufficient authority to bind these packages (BINDADD).

Configuring DB2 Connect EE, ASE or UE to Support Siebel Web Client

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

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

This approach is illustrated in the middle and lower parts of Figure 1 on page 25.

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 server, with protocol support for DB2 for z/OS.

   You can also add DB2 Connect to an existing server on which DB2 is already installed.

2. On the DB2 Connect machine, upgrade DB2 Connect to the appropriate Fix Pack. See "Installing Required IBM Fix Packs" on page 25 for further information.

3. On the DB2 Connect EE Server machine, catalog your DB2 for z/OS database, as appropriate, using the DB2 Client Configuration Assistant or the DB2 Command Line Processor.

   After installation, use a standard DB2 client to access the DB2 Connect EE Server.

Configuring DB2 Connect EE, ASE or UE to Support Siebel Developer Web Client

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

- A computer on which you install DB2 Connect EE, to act as a gateway to the database. This computer can be a Siebel Server machine or a separate machine.
- DB2 for z/OS.
- A Siebel Developer Web Client workstation.

This configuration enables a high volume of concurrent network transactions between the Siebel Developer Web Client and the DB2 Connect EE Server machine. This approach combines elements from the middle and lower configuration options shown in Figure 1 on page 25.

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 server, with protocol support for DB2 for z/OS.

2. On the DB2 Connect EE Server, upgrade DB2 Connect EE to the appropriate Fix Pack. See "Installing Required IBM Fix Packs" on page 25 for further information.

3. On the DB2 Connect EE Server machine, catalog your DB2 for z/OS database, as appropriate, using the DB2 Client Configuration Assistant or the DB2 Command Line Processor.

   After installation, use a standard DB2 client to access the DB2 Connect EE Server.

Configuring DB2 Connect PE to Support Siebel Developer Web Client

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

- A Siebel Developer Web Client workstation on which you install DB2 Connect Personal Edition (PE), to act as a gateway to the database.
- DB2 for z/OS.

With this approach, a user on a Siebel Developer Web Client connects directly to DB2. See the upper part of Figure 1 on page 25.

To connect to DB2 using DB2 Connect PE

1. Install DB2 Connect PE on the Siebel Developer Web Client workstation machine.
2. On the machine, upgrade DB2 Connect PE to the appropriate Fix Pack. See "Installing Required IBM Fix Packs" on page 25 for further information.

   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 z/OS databases as appropriate, using the DB2 Client Configuration Assistant or the DB2 Command Line Processor.

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

Additional Postinstallation Steps for DB2 Connect

After installing DB2 Connect, applying the necessary IBM Fix Pack, and setting up the network connectivity (TCP/IP and FTP) from the DB2 Connect server to the mainframe, there are a number of additional steps you must perform to use DB2 Connect effectively with Siebel applications. These steps include defining an alias, BINDing, and performing some additional configuration.

DB2 Connect configuration parameters allow you to specify some of the ways DB2 Connect works, for example, pooling and thread reuse. These configuration changes vary depending upon site-specific details and whether the Enterprise or Personal version of DB2 Connect is installed. Changes to these parameters change the database manager configuration file. For information on changing the database manager configuration parameters, refer to the appropriate IBM documentation.

Defining a Database Alias and Testing a Connection

This section describes how to define a database alias and test a connection.

When you create a connection to DB2 on z/OS, the connection is defined with a database alias that has the appropriate 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 TXNISOLATION 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 and for using the DB2 Configuration Assistant or Command Line Processor, refer to the relevant IBM documentation.

To test the database connection on the Windows platform

1. To ensure that you can connect to the database alias that you have defined, use either the DB2 Configuration Assistant or the DB2 Command Window or Command Line Processor.
2. To use the DB2 Configuration Assistant method, start the utility then highlight the database alias that you created, and right-click.

   A dialog box appears allowing you to select an option to test the connection.
3. Enter an authorized user ID and password, and select ODBC and CLI as the test parameters.

   The following message indicates a successful connection:

   ```
   CLI connection tested successfully.
   ODBC connection tested successfully.
   ```

Alternatively, you can use the DB2 Command Window or the DB2 Command Line Processor to test the database connection by typing the following and then pressing ENTER:

```
DB2 connect to dbalias user database userid using database password
```

where *dbalias* is the database alias you created using DB2 and *userid* and *password* are an authorized user ID and its associated password on the DB2 host.

If the connection is successful, database connection information appears.

To test the database connection on the UNIX platform

1. Open a UNIX shell.
2. Go to the *SIEBSRVR_ROOT* directory, that is, the directory in which the Siebel Server is installed, for example, home/siebel/siebsrvr.
3. Enter the following command:

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

   where *dbalias* is the database alias you created using DB2 and *user_ID* and *password* are an authorized user ID and its associated password on the DB2 host.

   If your connection is valid, database connection information appears.

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. 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 the relevant IBM documentation for further details on the BIND command.

Issue the BIND command at the time of installation and when subsequent IBM Fix Packs are applied. You issue the BIND command only once per subsystem. If you do not bind DB2 connect packages, SQL errors or authorization errors occur when a client without BINDADD authority attempts to connect and the DB2 software attempts to perform an autobind.

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 DB2 Subsystem

In setting up the DB2 subsystem in preparation for deploying a Siebel application on DB2 for z/OS, you must perform a number of tasks, and consider a number of factors, including the following:

- "Advantages of Using a Separate DB2 Subsystem" on page 31
- "About Unicode Conversion Tables" on page 32
- "Considerations in Choosing the Database CCSID" on page 32
- "The Data Distribution Facility and Workload Manager" on page 33
- "DDSNZPARM Parameter Settings for Siebel Applications" on page 34

Advantages of Using a Separate DB2 Subsystem

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

- **DSNZPARM optimization.** The DSNZPARMs used with Siebel Business 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 Business 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.** Siebel Business 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 an 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 7.7x approach of defining one table space per 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 separate DB2 subsystem is preferable for implementing Siebel Business Applications, there are numerous successful Siebel deployments in shared DB2 subsystems.

About Unicode Conversion Tables

Siebel Business 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/OS Unicode Conversion Services images do not include the required conversion tables.

You must configure the z/OS 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 that support a substantial number of codepage conversions.

Considerations in Choosing the Database CCSID

Siebel Systems supports both ASCII- and EBCDIC-based coded character set IDs (CCSIDs) on DB2 for z/OS. The database CCSID you use determines a number of factors, for example, the sort order used in list applets.

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 (for upgrades only)
- Compiling a new Siebel repository file (.srf file) from the new repository

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.

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."

NOTE: DB2 for z/OS v7 only supports a single code page for tables referenced within a single SQL statement.

Setting the CCSID

Configure the CCSIDs on the DB2 installation panel, DSNTIPF, in the fields ASCII CODED CHAR SET or EBCDIC CODED CHAR SET. The field DEF ENCODING SCHEME on the same panel determines whether or not the default encoding scheme is ASCII or EBCDIC.

NOTE: Set the value of the MIXED DATA field on the DSNTIPF panel to NO for all supported languages except Japanese.

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 Business 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/OS 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.

DDSNZPARM Parameter Settings for Siebel Applications

To run your Siebel Business applications, you must set the values of some of the DSNZPARM parameters to required settings; other DSNZPARM parameter values are recommended to improve the performance of your Siebel Business applications. This section describes:

- "Required DSNZPARM Parameter Settings" on page 34
- "Recommended DSNZPARM Parameter Settings" on page 35

NOTE: Use the IBM default values for any DSNZPARMs for which recommended or required values are not listed in this section.

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

Required DSNZPARM Parameter Settings

The DSNZPARM values that are required by Siebel Business Applications varies, depending on the version of DB2 for z/OS that you are using:

- If you are using DB2 for z/OS v7, the following parameter values are required:
  - CACHEDYN=YES
  - CDSSRDEF=1
  - EXTSEC=YES
  - PARTKEYU=YES
  - RETVLCFK=NO
  - MXTBJOIN=128
  - OPTCCOS1=ON
  - OPTSUBQ1=YES
- If you are using DB2 for z/OS v8, the following parameter values are required:
  - CACHEDYN=YES
  - CDSSRDEF=1
  - EXTSEC=YES

- PARTKEYU=YES
- RETVLCFK=NO

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

Recommended DSNZPARM Parameter Settings

For optimum operation of Siebel Business Applications, use the recommended settings in Table 7 to configure the parameters within DSNZPARM.

Table 7. Database Manager Configuration Parameters (DSNZPARM)

Parameter	Explanation	Applicable DB2 Version	Recommended Setting
DSN6SPRM			
CACHEDYN	Turns on dynamic statement caching.	v7 v8	YES (required)
CDSSRDEF	Turns off parallelism for dynamic statements.	v7 v8	1 (required)
CONTSTOR	Compresses storage on a regular basis. Set this value to YES if DBM1 storage is an issue.	v7 v8	YES
EVALUNC	Allows predicate evaluation on uncommitted data.	v7 v8	YES
IRLMRWT	Time in seconds before a time-out occurs.	v7	300
MAXKEEPD	The total number of prepared, dynamic SQL statements that can be saved past a commit point by applications that run with the KEEPDYNAMIC(YES) bind option.	v7 v8	0
MINSTOR	Compresses storage on a regular basis. Set this value to YES if DBM1 storage is an issue.	v7 v8	YES
MXTBJOIN	Sets the maximum number of tables in a join. The default (15) is not sufficient for Siebel Business Applications. **NOTE:** Unlike other DSNZPARMs that cannot be updated through the installation panels, this parameter must be updated manually.	v7 v8 (default)	128 (required)

Table 7. Database Manager Configuration Parameters (DSNZPARM)

Parameter	Explanation	Applicable DB2 Version	Recommended Setting
NPGTHRSH	Allows small tables to use indexes.	v7 v8	10
NUMLKUS	Number of locks per user. The DBA should monitor and set this value. If you experience persistent locking problems, consider setting the parameter to 0 (unlimited number of locks).	v7 v8	100,000
OPTCCOS1	Allows multiple, predicate optimizer enhancements.	v7 v8 (default)	ON (required)
OPTSUBQ1	Allows noncorrelated, subquery optimizer enhancements.	v7 v8	YES
PARTKEYU	Allows update of partitioning keys.	v7 v8	YES (required)
RETVLCFK	Allows index-only access of varying-length characters. This value should be set to NO due to padding.	v7 v8	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.	v7 v8	20
CONDBAT	Maximum number of concurrent remote connections. The DBA should monitor and set this value.	v7 v8 (default)	10,000
EXTSEC	Allows DB2 Connect to receive more complete error messages. Allows you to change passwords from DB2 Connect.	v7 v8 (default)	YES (required)
MAXDBAT	Maximum number of database threads (DBAT). The DBA should monitor and set this value.	v7 v8	500

Table 7. Database Manager Configuration Parameters (DSNZPARM)

Parameter	Explanation	Applicable DB2 Version	Recommended Setting
MGEXTSZ	Specifies whether or not the secondary extent allocations for DB2-managed data sets are sized according to a sliding scale that optimizes the likelihood of reaching the maximum data set size before secondary extents are exhausted. If you specify YES, DB2 automatically optimizes the secondary extent allocations.	v8	NO
DSN6FAC			
CMTSTAT	Allows a greater number of remote threads without affecting storage. Enables DDF thread pooling.	v7 v8 (default)	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.	v7 v8	600
POOLINAC	Number of seconds an inactive thread remains in the DDF pool. The DBA should monitor and set this value.	v7 v8 (default)	120
PADIX	Determines whether or not new indexes are padded by default. If you specify YES, a new index is padded unless the NOT PADDED option is specified on the CREATE INDEX statement.	v8	YES

NUMLKUS Considerations

If a resource unavailable error occurs because NUMLKUS has been exceeded while performing a Siebel operation, increase the NUMLKUS parameter value.

NOTE: The NUMLKUS 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.

If EIM fails because the value specified for the NUMLKUS parameter has been exceeded, take one of the following actions:

- 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.) Monitor this value in your installation to ensure that it is set properly. The IDTHTOIN parameter time-outs active threads that have been idle (no network traffic) for more than 600 seconds. (This time is the recommended value; your value may differ.)

The events that can occur when an active thread times out include:

- The thread is *not* returned to the DB2 Connect thread pool for reuse.
- When the client or component tries to reuse the DB2 Connect thread, the thread is not available. The client or component has to issue a full Connect to continue processing the work.
- If the component or client cannot issue a full Connect, the client receives a SQL30081N SQLCODE. An example of an error message you may receive if the component or client cannot reconnect is:

  ```
  [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 to bear in mind when specifying a value for the IDTHTOIN parameter is that connection pooling on the ODBC driver must be turned off. Using connection pooling with the driver is not certified with Siebel 7.x.

Estimating the Storage Space Required

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

The *minimum* DB2 storage space required for a typical Siebel 7.8 installation on DB2 for z/OS (installation of repository tables and seed data but with no user data loaded into the base tables) is between 5 GB and 20 GB, depending on the DEFINE parameter setting specified when the database objects were created.

If database objects are created with the DEFINE parameter set to YES, a DB2 catalog entry is created for the object and a data set is allocated for the object. If database objects are created with the DEFINE parameter set to NO, only a DB2 catalog entry is created for the object, so less space is required. Siebel Business Applications ship all table space and index schema definitions with the DEFINE NO option as the default. Therefore, 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.

The minimum values specified in this section vary depending on the Siebel functionality you implement and the amount and nature of data supporting it. The process for making accurate database size calculations is complex, involving many variables. Use the following guidelines to assist you:

- Determine the total number and types of users of Siebel Business 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.

NOTE: If you carry out a repository import and it fails, additional storage space is also required.

Allocating Space for Buffer Pools and Storage Groups

The following example illustrates how you can 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 you can 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 Business Applications. These tables and their indexes should be separated across devices. As a general rule, indexes should be on different physical devices to 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 Business Applications. Over 4,000 tables and 21,000 indexes are shipped for Siebel Industry Applications.

The number of objects created in the Siebel Schema is determined by the Siebel product line you purchase. The number of objects 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. However, if you need to drop unused, nonunique, Siebel indexes on the z/OS, contact Siebel Technical Support for assistance.

5 Configuring the Database Layout

This chapter describes the storage control file templates that are shipped with Siebel Business 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 Installation

The Siebel Database Server 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 Business Application. The DDL control file (ddl.ctl 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 for z/OS.

  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. Your 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.

  NOTE: It is especially important that you review SEGSIZE, PRIQTY, and SECQTY for the table spaces that are to 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 Business Applications with all the information necessary to create DDL for the Siebel Schema in an output file (by default named schema.sql).

You can choose either of two methods to execute the SQL files from a designated Siebel Server:

- Method 1. Automatic execution from the 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 24 to determine the configuration appropriate to the Client your enterprise supports, and then follow the instructions in "Installing Required IBM Fix Packs" on page 25.
- Method 2. Direct execution on the DB2 host using Siebel-provided scripts, your own FTP, or any other file transfer program to transfer files to the DB2 host, and then use a native IBM utility, such as SPUFI for file execution.

  NOTE: If you want to execute SQL using mainframe native tools, refer to the IBM documentation.

Siebel Storage Control File Templates

The storage control file is unique to the DB2 for z/OS platform for Siebel Business 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\db2390` directory (Windows) or the `/dbsrvr/db2390` directory (UNIX). Select one of the following templates, based on your partitioning needs, if you are using partitioning, and your encoding scheme:

- **storage_np.ctl**

  This template contains a database storage layout for the Siebel Schema with one table per table space, and one table space per database. No partitioned table spaces are provided.

- **storage_p.ctl**

  This template is the database storage layout for the 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 the 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; your DBA must customize them to suit the needs of your enterprise, for example, to change the default space allocation for Siebel objects, seed data, and views.

Although Siebel-provided templates might be sufficient for your development environment without any changes, your DBA should 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 how Siebel objects are grouped, and you should understand object naming conventions.

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 topics are described in this section:

- "Symbolic Variables in Storage Control Files" on page 45
- "Default Objects in Storage Control Files" on page 46
- "Buffer Pools Used in Storage Control Files" on page 47
- "About Using Storage Groups" on page 47
- "Database Objects in Storage Control Files" on page 47
- "Table Objects in Storage Control Files" on page 48
- "Table Space Objects in Storage Control Files" on page 49
- "Object Naming Conventions" on page 50.

Symbolic Variables in Storage Control Files

The storage control file contains definitions of objects (such as databases, table spaces, and tables) that define the physical storage layout of your database. Many of the options used in the object definitions 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 are substituted with the actual values that comply with your organization's standards. Review Table 9 for a list of the symbolic variables in the storage control file templates and the values substituted for them.

Table 9. Symbolic Variables

Symbolic Variable	Actual Value Substituted for Symbolic Variable
$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)

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   = SIDB0000
Tablespace = SIEBTS00
Stogroup   = SYSDEFLT
IndexStogroup= SYSDEFLT
Bufferpool = BP1
IndexBp    = BP2
SegSize    = 32
LockSize   = Page
LockMax    = 0
PriQty     = 48
SecQty     = 1440
Define     = No
PctFree    = 30
FreePage   = 4
Erase      = No
CCSID      = ASCII
Compress   = No
Type       = Defaults
```

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 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 "Process of Extending the Siebel Schema" on page 141.

Buffer Pools Used in Storage Control Files

Siebel Business 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 for a Custom Installation" on page 102 to enter these values.

Your DBA must 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 39 for an example of buffer pool definitions.

About Using Storage Groups

Siebel Business 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 39 for an example of storage group definitions.

Database Objects in Storage Control Files

In the Siebel 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 purchases. 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       = SIDB2532
LockSize   = Page
```

The Database object illustrated in this section translates into the following DDL output:

```
CREATE DATABASE SIDB2532 CCSID ASCII BUFFERPOOL BP1 /
```

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

LockSize is not 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, SIDB) 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 "Table Space Naming Conventions" on page 50.

Table Objects 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   = SIDB1465
Tablespace = H1465000
Clobs     = No
```

The Table object illustrated in this section 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 Business Applications include one template per encoding scheme when partitioning is used and one template per encoding scheme for nonpartitioning.

Most tables use page-level locking by default, but a few use row-level locking. Table 10 lists tables that use row-level locking.

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

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

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

Table Space Objects 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: You 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   = SIDB1004
LockSize   = Page
Bufferpool = BP1
Define     = No
Partitions = 0
```

This Tablespace object translates into the following DDL output:

```
CREATE TABLESPACE H1004000 IN SIDB1004
USING STOGROUP SYSDEFLT
PRIQTY   48 SECQTY 1440 FREEPAGE 4 PCTFREE 30
DEFINE NO
SEGSIZE 32
BUFFERPOOL BP1 LOCKSIZE 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. These additional options occur 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).

Object Naming Conventions

Understanding Siebel 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 data sets (VSAM files); for example, the name of the underlying data sets for the H0401000 table space might look like Q10E.DSNDBC.SIEBTO.H0401000.I0001.A001 where the fourth node represents the table space name.

NOTE: Siebel utilities can 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.

Table Space Naming Conventions

The table space naming convention applies to all Siebel table spaces, whether you create the table spaces or the table spaces are shipped as part of your Siebel Business Application.

Table spaces are named with eight characters (for example, H0401000), consisting 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 A four-digit number that is assigned to the base table and is stored in the GROUP_CD column of the S_TABLE table in the 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 table numbers are stored in the Group Code column (GROUP_CD) in S_TABLE. You can view Table and Group names in Siebel Tools under Table > Object.

CAUTION: You can edit the *Group* property for tables you create but you should not edit the *Group* property for tables provided in the Siebel application; doing so can cause operating anomalies in the application.

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

NOTE: 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. Using this prefix and suffix maintains object naming conventions.

Naming Conventions for Auxiliary Objects

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

Auxiliary object names comprise 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 (H)	X	A	I
SIA (V) 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.)

Naming Conventions for Partitioning Triggers

Partitioning Trigger names comprise 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 you can use to 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 ("Process of Performing a Custom Installation" on page 113) 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 you choose to use this method, you must 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, the Synchronize Schema Definition (ddlsync) and Migrate Repository (dev2prod) database configuration utilities.

  You can generate an updated version of the storage.ctl file using the Database Server Configuration Wizard. For further information, see "Extracting a Storage Control File from the DB2 Catalog" on page 103.

- **Method 2—Modify Generated DDL Using DB2 Tools**

  To use this method, perform a standard installation (see "Performing a Standard Installation" on page 85), and select the Generate DDL into Files installation mechanism. After the Siebel 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 this method, because it involves using customary DB2 tools and does not involve manipulation of the storage control file.

  When your DBA has modified the DDL, you should extract the storage control file so that it reflects the schema layout on the mainframe. This extraction saves time in the future as it means less manual work is involved when you run database utilities, such as Migrate Repository (dev2prod).

Modifying the Storage Control File

You can adjust the storage control file using either of the following methods:

- Open the storage control file with a text editor program and edit it directly to adjust object definitions or options.

  This approach is useful for making minor adjustments to the database layout but should only be used if you have a thorough understanding of the storage control file structure. Otherwise, you could introduce errors into the file and affect the operation of your Siebel applications.

- Use the Siebel Database Storage Configurator (dbconf.xls).

  This 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.

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

  The dbconf.xls file is installed in the *DBSRVR_ROOT*\db2390 (Windows) or $*DBSRVR_ROOT/*db2390 (UNIX) subdirectory. To open dbconf.xls, you must have Microsoft Excel installed on your workstation. For further information, see “Modifying Storage Control Files Using the Database Storage Configurator” on page 108.

  NOTE: If your installation is 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.

6 Understanding Siebel Partitioning

This chapter provides information on partitioning large tables in Siebel Business Applications. Partitioning a DB2 table is not a complicated procedure, but 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 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.

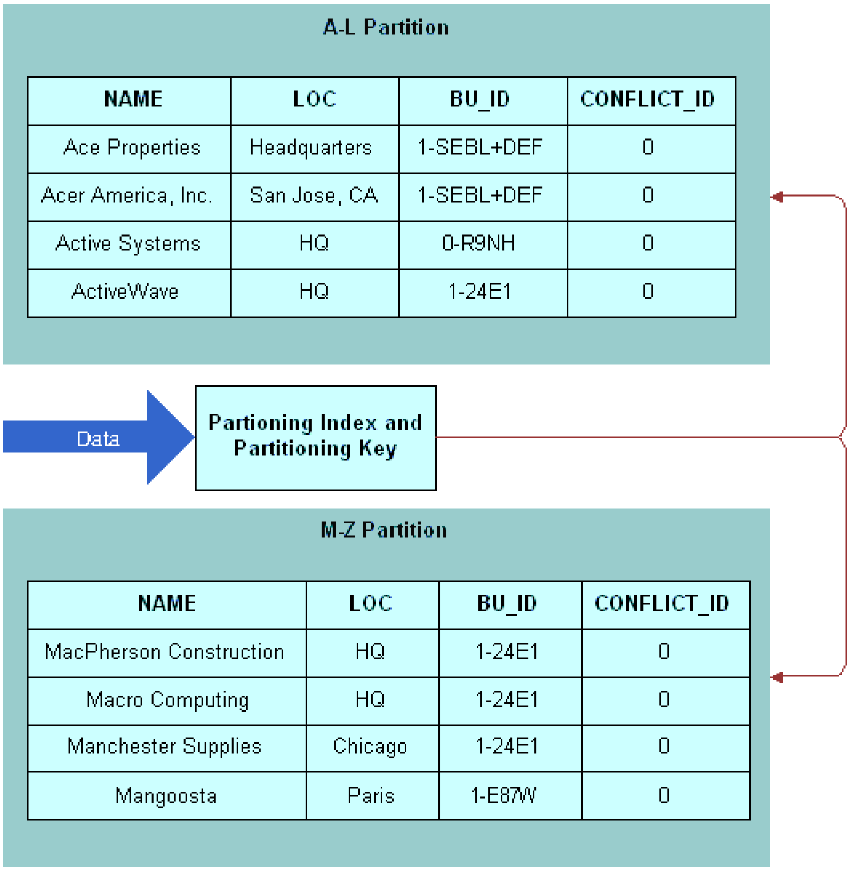
Any table, including any of the high-volume tables listed in Table 8 on page 40, can be partitioned during the installation or upgrade process. You can define partitioned table spaces and key ranges for Siebel tables during or after installation, based on your business requirements. For a complete list of prepartitioned Siebel tables, see "Prepartitioned Siebel Tables" on page 66.

You can 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 for z/OS. If you use the Siebel default partitions, you can either accept them as they are, or you can reconfigure them to suit your requirements.

About Partitioning Indexes and Keys

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.



NAME	LOC	BU_ID	CONFLICT_ID
Ace Properties	Headquarters	1-SEBL+DEF	0
Acer America, Inc.	San Jose, CA	1-SEBL+DEF	0
Active Systems	HQ	0-R9NH	0
ActiveWave	HQ	1-24E1	0

NAME	LOC	BU_ID	CONFLICT_ID
MacPherson Construction	HQ	1-24E1	0
Macro Computing	HQ	1-24E1	0
Manchester Supplies	Chicago	1-24E1	0
Mangoosta	Paris	1-E87W	0

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

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

Partitioning and the Storage Control Files

Partitioning is defined in Siebel storage control files. You can use either of the following storage control file templates shipped with your Siebel Business Applications to partition tables:

- **storage_p.ctl.** This template is the database storage layout for the Siebel Schema with partitioning for ASCII encoding.
- **storage_p_e.ctl.** This template is the database storage layout for the Siebel Schema with partitioning for EBCDIC encoding.

Storage control files are described in detail in Chapter 5, "Configuring the Database Layout."

Both partitioning storage control file templates contain the same Siebel-recommended partitioning schema. 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. The default partitioning scheme is optimized for online activities from the Siebel Client.

Siebel Business 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.

Considerations in Partitioning Tables

Careful planning, requirements analysis, and monitoring are necessary to achieve optimal partitioning. Decide which Siebel tables to partition and how to partition them on the basis of table size and usage in your deployment. You can partition any table in accordance with your business requirements; you can select all the tables that Siebel Systems provides for partitioning, or you can select a subset of them.

Choosing a Partitioning Key Column

You can use any column as the partitioning key for a Siebel table. In deciding which column is most appropriate for partitioning, consider data access and data distribution factors. You also need to determine the number of partitions your implementation requires.

NOTE: When choosing a partitioning key for a Siebel table, follow all the rules, restrictions, and concerns listed in the IBM DB2 documentation. Factors to consider include data distribution and potential updates of partitioning keys.
For example, when partitioning keys are updated, system performance is adversely affected. When choosing partitioning keys, you can avoid performance problems by selecting fields that are seldom or never updated.

Choosing 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 "Partitioning Strategies for Special Types of Tables" on page 63.

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

In this partitioning method, a table is partitioned based on existing columns in the Siebel Schema; for example, S_OPTY is partitioned by columns in the U1 index. 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 partitioning a table using an additional partitioning column, PARTITION_COLUMN, designed specifically for Siebel partitioning with even data distribution. 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.

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 the sequential order of ROW_ID, Siebel Systems provides a column, PARTITION_COLUMN, which 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 if the Siebel application is to recognize this column. You must also define this column as NOT NULL WITH DEFAULT.

The DB2 BEFORE INSERT trigger used to populate the PARTITION_COLUMN values is defined in the Siebel storage control file. This trigger extracts the last two bytes of a source _ID column and uses 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.

Example of Partitioning: the S_ADDR_ORG Table

This topic gives one example of partitioning a table and a table space. The example definitions in this topic reflect a partitioning scenario for the S_ADDR_ORG table, which resides in the H0401000 partitioned table space. The Siebel partitioning template used for this scenario is the storage_p.ctl storage control file.

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, depending on whether the table space is partitioned or nonpartitioned.

NOTE: The object numbers that identify the objects in this example might differ in your implementation.

Example of a Table Object Definition

The following is an example object definition for table S_ADDR_ORG:

```
[Object 6074]
Type       = Table
Name       = S_ADDR_ORG
Database   = SIDB0401
Tablespace = H0401000
Clobs      = No
```

Example of a Tablespace Definition

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

```
[Object 3197]
Type       = Tablespace
Name       = H0401000
Database   = SIDB0401
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 in 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 a partitioning index. The number of Tspart objects is equal to the number of partitions.

Example of a Tsparts Definition

In the example for the Tablespace object 3197, illustrated above, Partitions=10; therefore, ten Tspart objects must be defined, as shown in the following example for objects 3198 through 3208).

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

Example of the DDL for the Partitioned Table Space

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 SIDB0401 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
/
```

Example of an Index Definition

If you use the table S_ADDR_ORG, the column OU_ID appears to be a good candidate for defining 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. The new partitioning index S_ADDR_ORG_P99, containing PARTITION_COLUMN, is defined in the storage control file for S_ADDR_ORG.

A partitioning index definition in the storage control file requires two types of objects: IndexBase and IndexPart (index partitions). An example definition for each is shown.

IndexBase Definition

```
[Object 8386]
Type       = IndexBase
Name       = S_ADDR_ORG_P99
Table      = S_ADDR_ORG
Partitions = 10
SpecialCol = PARTITION_COLUMN WCHAR(2) NOT NULL WITH 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.

IndexPart Definition

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'
```

Example of the DDL for the Partitioning Index

The storage control file definitions for the partitioning index 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 144 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.

As shown in the previous example, the source column OU_ID is added as the second column of the index. Using this column order ensures that related records (records with the same OU_ID) are inserted and stored in physical clustered order. If the original clustering index contains more than one column, then these too can be added to the partitioning index to sustain clustering order. Schedule routine REORGs of the table space and index space regularly to sustain clustering order. Running REORGs regularly accommodates insert activity and reclaims PCTFREE and FREEPAGE definitions.

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

NOTE: 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 take this into account in defining the key ranges.

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 Business Applications provide generic values for the partitioning keys. 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.

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

Partitioning Strategies for Special Types of Tables

This section explains the special considerations you must take into account when partitioning tables that support specific Siebel business processes, such as Siebel Remote, Siebel Assignment Manager, Siebel Workflow, and Siebel Enterprise Integration Manager (EIM).

The topics in this section describe partitioning considerations for the following:

- "The Siebel Remote Transaction Logging Table"
- "The S_ESCL_REQ and S_ESCL_LOG Tables" on page 64
- "EIM Tables and Partitioning" on page 65.

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 Environments

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
SpecialCol = MEMBER_NAME WCHAR(8)
```

The S_ESCL_REQ and S_ESCL_LOG Tables

The 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.

You can partition these tables in one of the following ways:

- **Hard-code the actual values for GROUP_ID and RULE_ID.**

  This approach is recommended when the number of groups is static. You can define additional partitions to accommodate new values that are unknown at creation time. To use this option, first define the groups and then extract the GROUP_ID values you want to use for partitioning.

- **Add the PARTITION_COLUMN and a trigger to populate the column.**

  The number of partitions you define 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 resource contention occurs because multiple groups are using the same partition, consider increasing the number of partitions when you create new groups and assign new group IDs; this allows for a more granular level of GROUP_ID assignments to a partition range.

EIM Tables and Partitioning

To speed up Enterprise Integration Manager (EIM) load and reduce I/O (input and output) contention, spread partitions for EIM tables across the entire I/O subsystem. You can use the Siebel Database Storage Configurator to define partitions on EIM tables. For further information, see "Modifying the Storage Control File" on page 52.

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 the batch numbers used.

It is recommended that you partition EIM tables based on their U1 indexes, that is, IF_ROW_BATCH_NUM and ROW_ID. This method of partitioning allows an EIM batch input to be assigned to one partition, thereby allowing multiple EIM batches to be run in parallel.

To compute the optimal number of partitions for an EIM table, divide the number of rows in the EIM table by the number of parallel processes you intend to run. The result is the approximate number of partitions you should create for the table.

The recommended partitioning index is usually the U1 index (IF_ROW_BATCH_NUM and ROW_ID) but 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   )
```

EIM Table Partitioning and Data Distribution

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 *number*s.

If your storage control file is 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.

Prepartitioned Siebel Tables

Since release 7.7, Siebel Business Applications provide the following partitioned tables in storage control file templates. The database 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 ¬"

Table 12. Prepartitioned Siebel Tables with Physical Columns

Table	Index	Indexed Columns	Trigger Logic

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 ¬"

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 ¬"

Table 12. Prepartitioned Siebel Tables with Physical Columns

Table	Index	Indexed Columns	Trigger Logic

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

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_CVRG_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,INSCLAIM_NUM	"WHEN (N.INSCLAIM_NUM IS NOT NULL)
			BEGIN ATOMIC
			SET N.PARTITION_COLUMN = RIGHT(N.INSCLAIM_NUM, 2);
			END ¬"

S_INS_ITEM
Siebel Industry Applications (SIA) table
	S_INS_ITEM_P99	PARTITION_COLUM, 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

S_PARTY	S_PARTY_P99	PARTITION_COLUMN,ROW_ID	"WHEN (N.ROW_ID IS NOT NULL)
			BEGIN ATOMIC
			SET N.PARTITION_COLUMN = RIGHT(N.ROW_ID,2);
			END ¬"

Table 12. Prepartitioned Siebel Tables with Physical Columns

Table	Index	Indexed Columns	Trigger Logic

S_PARTY_REL	S_PARTY_REL_P99	PARTITION_COLUMN,PARTY_ID	"WHEN (N.PARTY_ID IS NOT NULL)
			BEGIN ATOMIC
			SET N.PARTITION_COLUMN = RIGHT(N.PARTY_ID,2);
			END ¬"

S_POSTN_CON	S_POSTN_CON_P99	PARTITION_COLUMN,POSTN_ID	"WHEN (N.POSTN_ID IS NOT NULL)
			BEGIN ATOMIC
			SET N.PARTITION_COLUMN = RIGHT(N.POSTN_ID,2);
			END ¬"

S_REVN	S_REVN_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_SRV_REQ	S_SRV_REQ_P99	PARTITION_COLUMN,SR_NUM	"WHEN (N.SR_NUM IS NOT NULL)
			BEGIN ATOMIC
			SET N.PARTITION_COLUMN = RIGHT(N.SR_NUM,2);
			END ¬"

7 Installing the Siebel Database Server Files at Midtier

This chapter describes how to install the Siebel Database Server software on both the Windows and UNIX platforms.

This chapter consists of the following topics:

- "About the Siebel Database Server"
- "Process of Installing the Database Server" on page 71
- "Installing the Database Server Software Under Windows" on page 73
- "Installing the Database Server Software Under UNIX" on page 75
- "Reviewing the Software Installation for DB2 for z/OS" on page 78

About the Siebel Database Server

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

Installation of the Siebel Database Server software provides files which are used in installing and configuring the Siebel Database Schema and Siebel Database Server components on the DB2 host and in importing the Siebel Repository.

Process of Installing the Database Server

To install the Siebel Database Server, perform the following tasks:

1. Fill out your copy of Appendix A, "Deployment Planning Worksheet" with the parameter values that you need to perform the installation. Distribute copies to the entire installation team. See Appendix A, "Deployment Planning Worksheet."
2. Ensure all the prerequisite installation tasks are completed. See "Preinstallation Tasks for the Siebel Database Server" on page 72.
3. Install the Siebel Database Server software. See "Installing the Database Server Software Under Windows" on page 73 and "Installing the Database Server Software Under UNIX" on page 75.
4. Review the software installation. See "Reviewing the Software Installation for DB2 for z/OS" on page 78.

5. Install and configure the Siebel Schema. See Chapter 8, "Installing the Siebel Schema on the DB2 Host."
6. Import the Siebel repository and complete post-installation tasks. See Chapter 9, "Siebel Repository Import and Postinstallation Tasks."

Preinstallation Tasks for the Siebel Database Server

Before installing the Siebel Database Server, make sure that you have the following resources available to you and have completed the following tasks:

- You have reviewed the *Siebel Installation Guide* for the operating system you are using.

  You have installed 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.

- You have designated a Siebel Server to act as a client to DB2. This server is where you 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.

- You have installed DB2 Connect, as described in Chapter 4, "Preparing for Implementation on the DB2 Host," and tested ODBC connectivity between the machine on which the Siebel Server is installed and DB2.
- 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 have created 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 on authorization requirements, see Chapter 3, "Security Concepts for z/OS."

- A database user (for example, SADMIN) with DBA/CREATEDBA authority exists. This user must also have TSO logon capability for the manual installation process.
- Your system administrator (SYSADM) and database administrator (DBA) have set up the DB2 subsystem. See the instructions in Chapter 4, "Preparing for Implementation on the DB2 Host."
- The SYSADM has created storage groups and activated and granted use of buffer pools in preparation for installation.
- The Database Owner is a valid authorization ID.
- (Optional) You have installed Microsoft Excel on the client machine to allow use of the Siebel Database Configurator tool.

CAUTION: You cannot install Siebel Business Applications unless you have between 50 and 100 MB of disk space on your system drive, even if you intend to install Siebel Business Applications on another drive.

Installing the Database Server Software Under Windows

Complete the following steps to install the Siebel Database Server software under Windows. You must have already created a Siebel 7.8 installation image on your network and have installed the Gateway Name Server and a Siebel Server on your computer. For further information on all of these tasks, see the *Siebel Installation Guide for Microsoft Windows: Servers, Mobile Web Clients, Tools*.

To install the Database Server software under Windows

1. In Windows Explorer, navigate to the Siebel image location for the current software version. Then navigate to the directory in which the Siebel Enterprise Server 7.8 software is located and double-click setup.exe.

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

2. Verify that no other programs are running, and then click Next to continue with the setup program.
3. When a screen displaying license information appears, accept the license terms, and then click Next to continue.
4. Continue the installation process, according to the prompts you receive:
   - If this is the first time you are installing the Siebel Enterprise Server software, you are prompted to select the directory into which you want to install the server. Proceed to Step 5.
   - If this is not the first time you are installing the Siebel Enterprise Server software, setup detects an existing version of this software on your system and allows you to either:
     - Install a new instance of the server software by selecting the None.Install a new instance option, and then clicking Next. Proceed to Step 5.
     - Install a new language pack to an existing installation instance, by selecting the instance, and then clicking Next. Proceed to Step 9 on page 75.
5. Specify where you want to install the Siebel Database Server software.
   - To install the Siebel Database Server software into the default directory, `C:\sea78`, click Next.
   - To install the Siebel Database Server software into another directory, use Browse to select the directory you recorded in Appendix A, “Deployment Planning Worksheet.”

     The directory name must not contain spaces, apostrophes, hyphens, or other special characters. Underscores, however, are allowed.

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

6 You are prompted to select the type of server you want to install. Select the Database Server option, and then click Next.

CAUTION: If you choose to install and configure all the components for which your company has a license at once, you cannot later uninstall any of them individually. You must uninstall them as a group.

If you install all licensed components at once, the Siebel Enterprise Server 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.

7 Choose the type of Siebel Database Server installation you want to perform from the following options and click Next.

- **Typical.** Select this option to install all Siebel Database Server components. Proceed to Step 9.
- **Compact.** There is no compact installation option for this server.
- **Custom.** Select this option to select the specific components you want to install. Proceed to Step 8.

8 Choose the database platform components you want to install from the Select Components screen. The following table shows the components available for you to install.

Installation Setup	Products
Typical or Custom	Select the database files you want to install from the following components: ■ Sample Database support—File attachments for Siebel seed data ■ Microsoft SQL Server 2000 ■ Oracle Database Enterprise Edition—Database scripts for the Siebel Database on Oracle ■ IBM DB2 for UNIX and Windows—Database scripts for the Siebel Database on DB2 for UNIX and Windows ■ IBM DB2 for z/OS —Database scripts for the Siebel Database on DB2 for z/OS

In this case, select the IBM DB2 for z/OS option and the Sample Database support option, and then click Next.

CAUTION: If you select a Custom installation, ensure you select the Sample Database support option the first time you install the software. You copy these sample file attachments to the Siebel File System directory after configuration.

9 Choose the Language Pack or Packs you want to install from the list displayed, and click Next.

Only language packs in the Siebel Enterprise Server directory are listed as install options. All servers are installed, at a minimum, with the primary (or base) language in which the server runs. Optionally, you can install one or more additional languages.

When you install 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 the *System Requirements and Supported Platforms* on Siebel SupportWeb.

10 The installer now prompts you that it will add program shortcuts to the Siebel Enterprise Server Configuration 7.8 Program Folder.

Click Next to continue or, to add the program shortcuts to a different program folder, first enter the name in the Program Folder field.

11 You are now prompted to verify the components you have chosen to install and their location. Either click Next to continue or Previous to amend your selections.

12 The language pack and Siebel Database Server software is now installed.

- If the installation is not successful, the Event Log appears, displaying the status of your installation. This log can also be accessed under sea78\error.log.
- When the InstallShield Wizard Complete screen indicates that software installation is completed successfully, click Finish.

Proceed to "Reviewing the Software Installation for DB2 for z/OS" on page 78.

Installing the Database Server Software Under UNIX

Complete the following steps to install the Siebel Database Server files under UNIX. You must have already created a Siebel 7.8 installation image on your network and have installed the Gateway Name Server and the Siebel Server on your computer. For further information on all of these tasks, see the *Siebel Installation Guide for UNIX: Servers, Mobile Web Clients, Tools*.

To install the Database Server software under UNIX

1 Navigate to the Siebel image location for the current software version, then navigate to the directory where the Siebel Enterprise Server 7.8 software is located, and open a new shell.

2 Verify that you have write and execute permission to the directories into which you want to install the Database Server software.

3 Unset any Siebel-specific environment variables.

To view current environment variable settings, enter env in the shell window.

4 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 the Siebel Enterprise Server screen appears.

5 Read the Welcome to InstallShield Wizard for Siebel Enterprise Server screen, and click Next.

6 A screen displaying license information is displayed. Accept the license terms, and then click Next to continue.

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 Database Server software, select the None.Install a new instance option, and click Next. Proceed to Step 8.
- To install a new language pack to an existing Database Server installation, select the displayed Database Server instance, and click Next. Proceed to Step 12 on page 77.

8 To install the Siebel Database Server software in the default directory, /sea78, click Next.

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

To install into a different installation directory, use Browse to select the directory you recorded in Appendix A, "Deployment Planning Worksheet."

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

9 When you are prompted to select the type of server that you want to install, select the Database Server option, and then click Next.

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.

If you install all licensed components at once, the Siebel Enterprise Server Installer and the Siebel Software Configuration Wizard 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.

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

- **Typical.** This setup option installs all Siebel Database Server components. If you chose the Typical installation type, proceed to Step 12.
- **Compact.** This setup option is not supported for this server.

- **Custom.** This setup option lets you customize your installation by choosing the specific components you want to install. Proceed to Step 11.

  The following table shows the components available for you to install.

Installation Setup	Products
Typical or Custom	Select the database files you want to install from the following components: ■ Microsoft SQL Server 2000 ■ Oracle Database Enterprise Edition—Database scripts for the Siebel Database on Oracle ■ IBM DB2 for UNIX and Windows—Database scripts for the Siebel Database on DB2 for UNIX and Windows ■ IBM DB2 for z/OS —Database scripts for the Siebel Database on DB2 for z/OS ■ Sample Database support—File attachments for Siebel seed data

CAUTION: If you select a Custom installation, ensure you select the Sample Database support option the first time you install the software. You copy these sample file attachments to the Siebel File System directory after configuration.

11 Select both the IBM DB2 for z/OS and the Sample Database support options, and then click Next.

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

Ensure that all the language packs you want to install are in the Siebel Enterprise Server directory. Otherwise, they are not listed as install options for you to select.

All servers are installed with the primary (or base) language in which the server is run. Optionally, you can also 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 the *System Requirements and Supported Platforms* on Siebel SupportWeb.

13 You are now prompted to verify the components you have chosen to install and their location. Either click Next to continue or Previous to amend your selections.

14 The language pack and Siebel Database Server software is now installed.

If the installation is not successful, the Event Log appears, displaying the status of your installation. This log can also be accessed under `sea78/error.log`.

15 When the InstallShield Wizard Complete screen indicates that software installation is completed successfully, click Finish.

Proceed to "Reviewing the Software Installation for DB2 for z/OS."

Reviewing the Software Installation for DB2 for z/OS

Review the directory structure created by the Siebel Database Server installation for your operating system, as illustrated in this section (this may vary depending on whether you performed a typical or a custom installation).

Windows Installation Directory for DB2 for z/OS

The db2390 subdirectory is located under the *DBSRVR_ROOT* directory within the Siebel root directory you specified during the installation, for example, `C:\sea78\dbsrvr\db2390\`.

DBSRVR_ROOT

- BIN
- COMMON
- DB2UDB
- DB2390
  - DBOUTPUT\
    - DDLSYNC
    - DEV2PROD
    - INSTALL
    - INSTALLDW
    - UPGRADE
  - STORPROC
  - UPGRADE
- *LANGUAGE* (for example, ENU)
  - DB2UDB
  - DB2390
  - MSSQL
  - ORACLE
- FILES
- LOCALE

MSSQL

ORACLE

PE60

- **BIN.** Contains files for internal use.
- **COMMON.** Contains common database platform-independent files.
- **DB2UDB.** Contains language-specific installation and upgrade files for DB2 UDB.
- **DB2390.** Contains language-specific installation and upgrade files for DB2 for z/OS.
  - **DBOUTPUT.** Contains subdirectories in which DDL is deposited by the installer for later transfer to the DB2 host when the Siebel Schema installation option, Generate DDL Into Files, is chosen.

    The subdirectories of 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 Generate DLL into Files option, 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.** Contains subdirectories containing files to allow upgrades from specific versions of Siebel Business Applications 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 132.
- **LOCALE.** Contains translation files for the Upgrade Wizard. For Siebel Systems use only.
- **MSSQL.** Contains language-specific installation and upgrade files for MSSQL.
- **ORACLE.** Contains language-specific installation and upgrade files for ORACLE.
- **PE60.** Files used for uploading data from Siebel Sales Personal Edition.

UNIX Installation Directory for DB2 for z/OS

Review the directory structure created by the Database Server installation for UNIX, as illustrated in this section (this may vary depending on whether you performed a typical or a custom installation).

```
$DBSRVR_ROOT
              bin
              common
              db2udb
              db2390
                          dboutput/
                                     ddlsync
                                     dev2prod
                                     install
                                     installdw
                                     upgrade
                          storproc
                          upgrade
              language
              (for example, enu)
                          db2udb
                          db2390
                          mssql
                          oracle
              files
              locale
              oracle
```

- **bin.** Contains files for internal use.
- **common.** Contains common database platform-independent files.
- **db2udb.** Contains language-specific installation and upgrade files for DB2 UDB.
- **db2390.** Contains installation and upgrade files specific to DB2 for z/OS.
  - **dboutput.** Contains subdirectories in which DDL is deposited by the installer for later transfer to the DB2 host when the Siebel Schema installation option, Generate DDL Into Files, is chosen.

    The following subdirectories 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 Generate DLL into Files option, 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.** Contains subdirectories containing files to enable upgrade from specific versions of Siebel Business Applications 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 may 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 132.

- **locale.** Contains translation files for the Upgrade Wizard. For Siebel Systems use only.

- **oracle.** Contains language-specific installation and upgrade files for ORACLE.

8 Installing the Siebel Schema on the DB2 Host

This chapter describes how to carry out a standard and a custom Siebel Schema installation and provides instructions for creating a storage control file for a custom installation.

This chapter consists of the following topics:

Process of Installing the Siebel Schema

After installing the Database Server software at midtier, you must install the Siebel Schema on the DB2 host using the Database Server Configuration utility.

To install the Siebel Schema on the DB2 host, perform the following tasks:

1. Review information about installing the tables, indexes, and seed data, using either the standard or custom option. See "About Installing the Siebel Schema."
2. Install the Siebel Schema:
   - If installing in standard mode, see "Performing a Standard Installation" on page 85.
   - If installing in custom mode, see "Preparing for a Custom Installation" on page 102.

     Perform the custom installation using a storage control file you have created. See "Process of Performing a Custom Installation" on page 113.
3. Review the log files for any errors. See "About Reviewing the Log Files for Windows" on page 115 or "About Reviewing the Log Files for UNIX" on page 117.

About Installing the Siebel Schema

When you choose the Install Database option on the Database Server Configuration utility to install the Siebel Schema, 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.

Installation Modes

You can create the Siebel Schema using one of two modes—standard or customized.

- **Standard Install**

  If you choose the standard installation option, you perform all installation tasks using the Database Server Configuration utility. A standard install creates a Siebel preconfigured database layout.

- **Customized Install**

  If you choose the custom installation option, you provide a storage control file, which you create and validate prior to performing the installation using the Database Server Configuration utility, to create the custom database layout that is most suitable for your Siebel deployment.

Installation Mechanisms

You can choose whether the DDL used to create the Siebel Schema is automatically applied against the DB2 host or is applied manually. You can execute the DDL directly against the database by choosing the Run DDL Automatically option. Alternatively, by choosing the Generate DDL Into Files option, you can assign the 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.

NOTE: The Upgrade Wizard is the Siebel utility used for database server installation, upgrade, and database manipulations. The utility is launched by the Database Server Configuration utility. In Windows, the Siebel Upgrade Wizard is designated as siebupg.exe. In UNIX, it is designated as srvrupgwiz and is located in the *$SIEBSRVR_ROOT/bin* directory.

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 utility and choosing Siebel Database Server Options > Install Database> Add a language to an existing database.

CAUTION: If you add a new language *after* completing your initial installation, when the Database Server Configuration wizard launches and you are prompted to enter the Gateway Name Server Address and the Enterprise Server Name, leave these fields blank. If you enter values in these fields, an error occurs.

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 the first step of a custom installation, and then customize your deployment after you have completed the standard installation. To perform a custom installation, see "Process of Performing a Custom Installation" on page 113.

The standard installation provides two schema layout options: Siebel Schema without Partitioning and Siebel Schema with Partitioning. As shown in Table 13, the storage control file you use during the standard installation is determined by your preferred Siebel Schema layout and your encoding scheme.

Table 13. Siebel Schema Layout and Storage Control File Templates

Siebel Schema Layout	Storage Control File Template to Use	Storage Control File Description
Siebel Schema without Partitioning	storage_np.ctl	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.
Siebel Schema with Partitioning	storage_p.ctl	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.ctl	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 "Siebel Storage Control File Templates" on page 44 for descriptions of the templates.

If you choose the Siebel partitioned layout, a selected subset of Siebel tables is defined as partitioned with predefined partitioning indexes. For a discussion of partitioned table spaces and partitioning indexes, see Chapter 6, "Understanding Siebel Partitioning."

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 7.8 > 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 or an error occurs.

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 value is the *SIEBSRVR_ROOT* directory, for example, `C:\sea78\siebsrvr`.)
   - Select Browse to select an alternate directory path.

   To continue, click Next.

4. On the Siebel Database Server Directory screen, either:

   - Accept the default path displayed in the Siebel Database Server Directory field. (This path is the *DBSRVR_ROOT* directory, for example, `C:\sea78\dbsrvr`.)
   - Select 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, IBM DB2 for z/OS, and then click Next.

6. On the Siebel Database Operation screen, select Install Database, and then click Next.

7. On the Select Installation Operation screen:

   - If you are configuring Siebel Database Server software for the first time, select Install Siebel Database, click Next, and then proceed to Step 8.
   - If you have already installed your Siebel Database Server software in its base or primary language, and want to install a new language, see "Installing Multilingual Seed Data" on page 127 for further instructions.
   - Install Siebel Data Warehouse Database. For information on this installation option, see the *Siebel Data Warehouse Installation and Administration Guide*.

   To continue, click Next.

8 On the Language Selection screen, select the language you want to use as the base language for the database, and click Next.

9 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; this database alias must be registered within DB2 Connect.

  NOTE: If you define database aliases, you must add the `TXNISOLATION` 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 to the one currently pointed to, you can type in the ODBC name of that Database Server, provided 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 you can use another ODBC data source name to generate a new Siebel Schema DDL.

- Enter the name of the source database DB2 subsystem.

To continue, click Next.

10 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.

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

- Siebel Schema Qualifier. Enter the character ID that identifies the Siebel Schema owner, for example, SIEBTO.

  NOTE: This ID can be up to eight characters in length, must be in uppercase, must start with a letter, and cannot include any special characters. This value 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."

To continue, click Next.

12 On the Select Installation Type screen, choose whether to perform a standard or a customized install, and then click Next.

- 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 provide a custom storage control file, which you have created and validated prior to performing this installation. This option allows you to create the database layout that is best suited to your Siebel deployment.

If you choose to do a Customized Install, See "Process of Performing a Custom Installation" on page 113 for information on the remaining steps in the custom installation process.

If you choose to do a Standard installation, proceed to Step 13.

13 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."

14 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 in segmented table spaces.
- Siebel Schema with Partitioning. This layout includes a set of tables that are recommended for partitioning. The remaining nonpartitioned tables are in segmented table spaces. For more information on this option, see Chapter 6, "Understanding Siebel Partitioning."

For details about layout considerations, see Chapter 5, "Configuring the Database Layout."

To continue, click Next.

15 On the Storage Group for Table Spaces screen, enter values for the following parameters:

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

To continue, click Next.

16 On the 4KB Buffer Pool Name screen, specify 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.

17 On the 32KB Buffer Pool Name screen, specify values for the following parameters:

- 32KB Buffer Pool Name. Enter 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.

18 On the Database Name Prefix screen, type the prefix for your database name. The default value 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.

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

- Generate DDL Into Files. Select this option to generate the DDL required to create the Siebel Schema into files. Click Next and proceed to Step 20.

  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 z/OS where the DBA executes it using customary methods, for example, SPUFI or DSNTEP2.

- Run DDL Automatically. Select this option to apply the DDL and DML required to create the Siebel Schema directly against the database, using ODBC to connect. Click Next and then proceed to Step 22.

20 On the DDL Commit Frequency screen, choose the number of DDL statements that can be run before a COMMIT statement is issued. Click Next.

21 On the Output Directory screen, indicate the output directory that is to contain schema.sql, ddlview.sql, and the new files to apply DDL on the mainframe, or accept the default directory, `C:\sea78\dbsrvr\db2390\dboutput\install`.

22 On the Log Output Directory screen, indicate the directory where you want to generate the log files.

Accept the default value, `install_mf`, or enter your new value. Click Next.

23 In the runnow message box, choose from one of the following options:

- To apply the configuration now, click OK. Proceed to Step 24.

- To apply the configuration later, click Cancel. You can apply or restart the configuration later by entering the following command from the command line:

  ```
  C:\sea78\siebsrvr\bin\siebupg.exe /m master_install_mf.ucf
  ```

  NOTE: This command is generated dynamically; the path name will vary based on the location of your Siebel Server installation.

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

- To change any of the values, click Previous to return to the screen with the parameter you need to change.

- If the values are correct, click Finish and the configuration is applied.

25 When you receive a message stating that the configuration was applied successfully, click OK.

This step completes the Run DDL Automatically installation option.

If you chose the Generate DDL Into Files option, you must complete the steps in the following section to complete the installation of the Siebel Schema from the generated DDL.

To Complete the Siebel Schema Installation Using Generated DDL

1 After the Siebel Upgrade Wizard generates the schema.sql and ddlview.sql files that contain the DDL to create the Siebel Schema, the following message appears:

```
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, please start
the Upgrade Wizard from the command line using option /m master_install_mf.ucf).
```

- Click No to quit the Siebel Upgrade Wizard until your DBA has had an opportunity to apply the Siebel Schema on the DB2 host. You can relaunch the wizard at any time to continue with this step. Proceed to Step 2.
- Click Yes if your DBA has already created the Siebel Schema on the DB2 host. The schema is now validated. Proceed to Step 4.

2 Transfer the schema.sql and ddlview.sql files to z/OS, 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 96.

The GRANT VIEW statements in ddlview.sql might cause SQL errors to be generated. For information on using GRANT VIEW statements in DB2 for z/OS v8, see "Granting Authorization to Views in DB2 for z/OS v8" on page 18.

3 Now that you have created the Siebel Schema on the DB2 host, restart the Siebel Upgrade Wizard from the command line as described in Step 23 on page 89.

The Pause #1 message shown in Step 1 is again displayed.

4 Click Yes in response to this message to validate the schema that has been created on the DB2 host.

When the schema has been successfully validated, the following message is displayed:

```
Pause #2: Please ensure that the schemvld.sql is empty, apart from trigger
creations, (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 option /
m master_install_mf.ucf).
```

Make sure that the schemvld.sql file is empty. This file is generated during the installation and placed in the default output directory (`dbsrvr\db2390\dboutput\install`) or into the output directory that you designated.

5 Indicate one of the following:

- Click No to discontinue the Siebel Schema installation process until later. Proceed to Step 6.

- If your DBA has already deployed user defined functions (UDFs) on the DB2 host, you can continue with the installation by clicking Yes. Proceed to Step 8.

6 Your DBA must now install user-defined functions on the DB2 host. See the Deploying User-Defined Functions section in "Applying DDL on the DB2 Host" on page 96 for further information.

7 Restart the Upgrade Wizard to continue the schema installation process.

8 The Siebel Upgrade Wizard displays the final step in the installation, that is, importing Siebel seed data.

  - Click OK and the Siebel Schema is populated with seed data.

    NOTE: This operation fails if the Siebel Schema has not been created because there is no schema to populate.

  - If you want to exit from the Upgrade Wizard, click Cancel. You can relaunch the wizard at any time to continue with this step.

  When you have successfully populated the schema with seed data, you are ready to carry out the final step in the Siebel Schema installation process - importing the Siebel Repository. For further information on this task, see "Importing the Siebel Repository" on page 119.

The following files are generated by the installation into the default output directory (`C:\sea78\dbsrvr\db2390\dboutput\install`) or into the output directory that you designated in Step 21 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.

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

Performing a Standard Installation Under UNIX

Perform the following steps to carry out a standard installation under UNIX.

To perform a standard installation under UNIX

1 Navigate to the Siebel Server root directory, for example, home/siebel/siebsrvr, and source environment variables, using one of the following commands, according to 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
    ```

2 Verify the values of the $*SIEBEL_ROOT* and *LANGUAGE* variables by entering the following:

```
env
```

The *LANGUAGE* variable should be set to the language in which the Database Server Configuration utility prompts appear, for example, enu for U.S. English.

The *SIEBEL_ROOT* variable should be set to the path of your Siebel Server installation directory, for example, home/siebel/siebsrvr.

3 To reset the environment variables, use 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
  ```

4 Navigate to $*SIEBEL_ROOT* /bin and type the following to launch the Database Server Configuration utility:

```
dbsrvr_config.ksh
```

The script displays the values of the *SIEBEL_ROOT* and *LANGUAGE* environment variables.

5 Review the values of these variables. If they are correct, enter Y and proceed to Step 6.

If the environment variable values are not correct, enter N and correct them before proceeding.

6 Specify the path of your Siebel Server root directory or press ENTER to accept the default path, /home/siebel/siebsrvr.

7 Specify the path of your Siebel Database Server root directory, or press ENTER to accept the default path, for example, /home/siebel/dbsrvr.

8 Select your RDBMS platform. In this case, select option 2 - IBM DB2 for z/OS.

9 Select option 1- Install Database on the Siebel Database Operation menu.

10 Select option 1- Install Siebel Database on the Installation Options menu.

- For installation of an additional database language, see "Installing Multilingual Seed Data" on page 127.
- For installation instructions for the Siebel Data Warehouse database, see the *Siebel Data Warehouse Installation and Administration Guide*.

11 Select the language in which you want to run your database by entering the corresponding number.

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

- Enter the default name of the ODBC data source, SiebSrvr_Siebel, or enter the database alias you prefer to use for the data source; you should have registered this alias within DB2 Connect.

  NOTE: If you want to work with information on a Database Server other than the one currently pointed to, you can specify the ODBC name of that Database Server, provided you have 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*.

- Enter the name of the source database DB2 subsystem, for example, Q20A.

13 Specify the Database User Name and the associated Database Password.

The Database User name is the user ID with authorization to log into the Siebel database. This user ID (for example, SADMIN) should have the necessary authorization to set the CURRENT SQLID.

The Database Server Configuration utility now verifies the ODBC and password values you entered. If these are invalid, you cannot continue with the Siebel Schema installation procedure. Either exit from the application by entering Y, or enter N to re-enter these values.

14 Specify the Siebel Schema Qualifier, for example, SIEBTO, or press ENTER to accept the default.

This is the identifier that designates the Siebel Schema and is also an authorization ID. The Schema Qualifier must be in uppercase, can be up to eight characters in length, must start with a letter, and cannot contain special characters.

15 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 Specify the type of installation you want to perform. In this case, select option 1 - Standard Install.

- 1 - Standard Install. The standard installation option installs the database using one of the Siebel-recommended default storage layouts. Proceed to Step 17.

- 2 - Customized Install. For a customized installation, you must provide a storage control file, which you create and validate prior to performing the installation. Using a customized storage control file allows you to create the database layout most suited to your Siebel deployment.

  See “Process of Performing a Custom Installation” on page 113 for information on the remaining steps in the custom installation process.

17 Specify the code page encoding scheme for your Siebel Database, either ASCII or EBCDIC, by entering the corresponding number.

For more information on choosing the database code page, see “Considerations in Choosing the Database CCSID” on page 32.

18 Specify the Siebel Schema layout you prefer, that is, with or without partitioning.

For details about layout considerations, see Chapter 5, “Configuring the Database Layout.”

19 Enter the name of the storage group for table spaces, or accept the default name, SYSDEFLT.

Discuss the name to use with the DBA before you start the installation.

20 Enter the name of the storage group for indexes, or accept the default name, SYSDEFLT.

21 Enter the name of your 4-KB buffer pool, or accept the default BP1.

22 Enter the name of your 16-KB buffer pool, or accept the default, BP16K1.

23 Enter the name of your 32-KB buffer pool, or accept the default, BP32K1.

24 Enter the name of your index buffer pool, or accept the default, BP2.

NOTE: The DBA should activate and grant these buffer pools prior to the installation.

25 Enter the Database Name Prefix. The default prefix is SIDB.

The prefix must consist of up to four of the first characters in the names of your logical Siebel databases, must start with a letter, and cannot contain any special characters.

26 Specify the mechanism you want to use for installation:

- 1 - Generate DDL Into a File. Generates the DDL required to create the Siebel Schema into files.

  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. Proceed to Step 27.

- 2 - Run DDL Automatically. Select this option 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 29.

27 On the DDL Commit Frequency screen, choose the number of DDL statements that can occur before a COMMIT is issued.

28 Specify the path of the output directory where you want the DDL files to be created; the default directory is `$`*`SIEBEL_ROOT`*`/dbsrvr/db2390/dboutput/install`.

29 Indicate the directory where you want the log files to be generated; the default directory is install_mf.

30 Review the configuration values you entered for accuracy.

- If the values are correct, launch the Siebel Upgrade Wizard (srvrupgwiz) by entering 1 (Yes).
- To change any of the values, enter 2 (No) and rerun the Database Server Configuration utility by entering `dbsrvr_config.ksh`; this allows you to reconfigure with different values.

When you receive a message that the configuration was applied successfully, this completes the installation option to Run DDL Automatically.

If you chose option 1 - Generate DDL Into a File in Step 26, the DDL is created and saved to a file but is not executed. You must complete the following steps to complete the installation of the Siebel Schema from the generated DDL.

To Complete the Siebel Schema Installation Using Generated DDL

1 If you chose the Generate DDL Into Files installation option in Step 26, the utility displays the following message after it generates the schema.sql and ddlview.sql files that contain the DDL to create the Siebel Schema (see Step 30):

```
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, please start
the Upgrade Wizard from the command line using option /m master_install_mf.ucf).
(Y/N)
```

- Enter N to quit the Siebel Upgrade Wizard until your DBA has applied the Siebel Schema on the DB2 host. Proceed to Step 2.
- Enter Y if your DBA has already created the Siebel Schema on the DB2 host, and you want to validate the installation. Proceed to Step 4.

2 Transfer the schema.sql and ddlview.sql files to z/OS and have your DBA apply them to create the Siebel Schema.

For information on this process, see "Applying DDL on the DB2 Host" on page 96.

3 When the Siebel Schema has been created on the DB2 host, restart the Siebel Upgrade Wizard. The Pause #1: message shown in Step 1 is again displayed.

4 Enter Y to validate the schema that you created on the DB2 host.

When the schema is successfully validated, the following message is displayed:

```
Pause #2: Please ensure that the schemvld.sql is empty apart from trigger
creations (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 option /m
master_install_mf.ucf).
(Y/N)
```

Make sure that the schemvld.sql file, generated during the installation and placed in the default output directory (dbsrvr/db2390/dboutput/install) or into the output directory that you designated, is empty.

5 Indicate one of the following:

- Enter N to discontinue the Siebel Schema installation process. Your DBA must now install user-defined functions on the DB2 host. For further information, see the Deploying User-Defined Functions section in "Applying DDL on the DB2 Host" on page 96.

  Proceed to Step 6.

- If your DBA has already deployed UDFs on the DB2 host, you can continue with the installation by entering Y to populate the Siebel Schema with seed data.

  NOTE: This operation fails if you have not created the Siebel Schema.

  Proceed to Step 7.

6 When you have deployed UDFs on the DB2 host, restart the Upgrade Wizard and enter Y to populate the Siebel Schema with seed data.

7 When you have successfully populated the schema with seed data, you are ready to carry out the final step in the Siebel Schema installation process - importing the Siebel Repository. For further information on this task, see "Importing the Siebel Repository" on page 119.

The following files are generated by the schema installation into the default output directory (`dbsrvr/db2390/dboutput/install`) or into the output directory that you designated in Step 28 on page 94.

- 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.

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.

Applying DDL on the DB2 Host

When you install the Siebel Schema, if you select the Generate DDL Into Files option, the schema.sql and ddlview.sql files are generated in the DDL output directory that you specified during the installation. Before continuing with the installation, you must transfer the DDL files to the DB2 host and then apply them. Both procedures are described in this section (the instructions in this section apply to both Windows and UNIX operating systems).

NOTE: Ensure that the tool you use (SPUFI or DSNTEP2) to apply the DDL on the DB2 host uses a slash as the SQL delimiter. See "Setting Up the SQL Delimiter on DB2" on page 100 for further information.

To transfer the DDL files to the DB2 host

1. Navigate to the output directory (Windows: `dbsrvr\db2390\dboutput\install`; UNIX: `dbsrvr/db2390/dboutput/install`) 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 all occurrences of *SIEBELQ1* to your own high-level qualifier (HLQ), for example, SADMIN. Save the file. The 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 &directoryPath1/jobinstl.txt 'SIEBELQ1.SIEBEL.INST.JOBINSTL'
send &directoryPath1/ddlview.sql 'SIEBELQ1.SIEBEL.INST.DDLVIEW'

quote site cylinders primary=10 secondary=25
```

```
send &directoryPath1/schema.sql 'SIEBELQ1.SIEBEL.INST.SCHEMA'

Quit
```

NOTE: The *&directoryPath1* variable is substituted with the appropriate UNIX or Windows path.

3. Save the changes you have made to instftp.txt.

4. Double-click instftp.bat from Windows or issue the following command from UNIX:

   ```
   ftp -vn < instftp.txt
   ```

5. Enter the password associated with the user name you entered in the instftp.txt file in Step 2 on page 96.

   Press any key to continue.

6. The files are sent to the DB2 host and the log file instftp.log is created in your *DBSRVR_ROOT*\db2390\dboutput\install directory (Windows) or your $*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 Quit command received. Goodbye.
   ```

You can now apply the DDL on the DB2 host.

To apply the DDL files on the DB2 host

1. When the DDL file transfer is successfully completed, log on to the mainframe and navigate to the *HLQ*.SIEBEL.INST.JOBINSTL data set, for example, SIEBLQ1.SIEBEL.INST.JOBINSTL.

   The DSLIST screen reflects the data sets associated with the user ID you entered previously.

2. Edit the *HLQ*.SIEBEL.INST.JOBINSTL data set by entering e to the left of the data set name and pressing ENTER.

   The Edit Entry Panel appears.

3. As instructed 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.

4. Enter submit on the Command line, and then press ENTER to submit the job.

   This job creates a new data set named *HLQ*.SIEBEL.INST.EXEC.

5. On the DSLIST screen, edit the *HLQ*.SIEBEL.INST.EXEC data set by entering e to the left of the data set name and pressing ENTER.

6. Enter exec to the left of the SBLINST member to start the installation.

7. On the Install Main Menu screen, you can select one of 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 the JCL and DDL used to generate the schema.

8. Enter the DDL input data set name, for example, *HLQ*.SIEBEL.INST.SCHEMA, and then press ENTER.

   An edit list appears for an SQL data set named *HLQ*.SIEBEL.INST.SCHEMA.SCH.

9. Press F3 to exit after the job completes.

10. Enter e to the left of the *HLQ*.SIEBEL.INST.SCHEMA.SCH data set, and press ENTER.

    NOTE: If you use a tool to create objects other than the DSNTEP2 tool, specify the program and plan names of the tool you are using in the SIEBSQLA member, for example:

    ```
    DSN SYSTEM(Q109)
    RUN PROGRAM(MYTEP2) PLAN(MYTEP2) PARMS(‘ /SQLTERM(/) ‘)
    END
    ```

11. In the *HLQ*.SIEBEL.INST.SCHEMA.SCH data set, edit the following JCL members in the order shown. Run the GRJCL member last because it grants permissions to the objects (database, table spaces and tables) created when you run the other three members.

    - DBJCL
    - TSJCL
    - TBJCL
    - GRJCL

    Edit and run each JCL member in turn as follows:

    a. Enter *e* to the left of the JCL member and press ENTER.

b Edit the job card parameters in each member, for example:

```
// D2PUNL0 JOB ACCNT#,CLASS=S,MSGCLASS=X,MSGLEVEL=(1,1),
// TIME=1440,REGION=0M,NOTIFY=&SYSUID,LINES=9999
//*TYPRUN=SCAN
```

c Enter an asterisk (*) to the left of the TYPRUN parameter to comment out this command.

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 occurred and that the job ran successfully.

The Siebel DDL is now generated.

12 To generate the View DDL file, follow the instruction to edit the *HLQ*.SIEBEL.INST.EXEC data set described in Step 5 on page 98.

13 Select 2 on the Install Main Menu screen to generate the View DDL.

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

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

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

NOTE: If you use a tool to create objects other than the DSNTEP2 tool, specify the program and plan names of the tool you are using in the SIEBSQLA member, for example:

```
DSN SYSTEM(Q109)
RUN PROGRAM(MYTEP2) PLAN(MYTEP2) PARMS(‘ /SQLTERM(/) ‘)
END
```

17 In the *HLQ*.SIEBEL.INST.DDLVIEW.VIE data set, edit the VIEJCL member:

a Enter e to the left of the VIEJCL member, and press ENTER.

b Edit job card parameters in each member, for example:

```
// D2PUNL0 JOB ACCNT#,CLASS=S,MSGCLASS=X,MSGLEVEL=(1,1),
// TIME=1440,REGION=0M,NOTIFY=&SYSUID,LINES=9999
//* TYPRUN=SCAN
```

c Enter an asterisk (*) to the left of the TYPRUN parameter to comment out this statement.

d Enter an asterisk (*) to the left of the GRANT statements; GRANT statements on views are not required with DB2 for z/OS. If you do not comment out these statements, error messages are returned when you run the VIEJCL member; you can ignore these messages.

e 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 occurred and that the job ran successfully.

18 Return to the midtier to continue the Siebel Schema installation by verifying the schema that you have just created.

NOTE: The schema.sql file is overwritten during the schema validation process. Therefore, if you would like to save the schema.sql file with the complete Siebel Schema DDL that was generated for the Siebel Schema, rename schema.sql (located in the *DBSRVR_ROOT*\db2390\dboutput\install directory on Windows or *$DBSRVR_ROOT*/db2390/dboutput/install directory on UNIX) after you apply schema.sql on the mainframe.

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 Business 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, you can replace slashes with semicolons.

DSNTEP2

If you run the DDL in batch mode, use the DSNTEP2 tool.

To set the delimiter in DSNTEP2

- Use the SQLTERM(/) parameter to set the delimiter to a slash mark. The following example illustrates how this appears in the SYSTSIN DD statement.

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

  ```
  //SYSTSIN  DD *

  DSN SYSTEM(Q202)

   RUN  PROGRAM(DSNTEP2) PLAN(DSNTEP71) PARMS('/SQLTERM(/)') -

        LIB('DSN710.RUNLIB.LOAD')

  END
  ```

SPUFI

If you use SPUFI, perform the following procedure.

To set the SQL delimiter in SPUFI

- Set CHANGE DEFAULTS to YES.

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

If you are not using triggers, you can change the slashes back to semicolons. If you use the TSO editor, do this by entering the following:

```
CHANGE X’61’ X’5E’ ALL
```

Validating Siebel Schema Creation

The second pause in the Siebel Schema installation process generates the schemvld.sql file. The schemvld.sql file contains SQL statements for missing Siebel Schema objects, if such objects are found. Normally, schemvld.sql is empty.

If schemvld.sql is empty, this means the Siebel Schema was created successfully. If the file is not empty, the Siebel Schema was not created successfully, probably because the DBA missed a step during the manual schema creation process on the DB2 host. The file contains the missing objects of the schema (the *Delta* schema). The DBA must apply the file with the missing objects to complete a full schema creation.

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

Deploying User-Defined Functions

Two user defined functions (UDFs) are provided in the Siebel application: nextseq and exrate. The nextseq() UDF generates sequential values for Enterprise Integration manager (EIM) export and the exrate() UDF is used for currency aggregation functions. After the schema is validated, you must install the UDFs using the following instructions. These instructions are applicable to both Windows and UNIX operating systems.

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 output directory (`*DBSRVR_ROOT*\db2390\dboutput\install` on Windows or `*$DBSRVR_ROOT*/db2390/dboutput/install` on UNIX) and modify the udfftp.txt FTP file as follows:

   a. Change the *&IP* address to the IP address or domain name of your mainframe.

   b. Change the *&username* to a valid TSO user ID.

   c. Change the *SIEBELQ1* high level DSN qualifier to your DSN qualifier.

   d. Save the changes to the file.

2. Double-click *DBSRVR_ROOT*\db2390\dboutput\install\udfftp.bat (Windows) to execute the batch FTP script to send files to the mainframe, or issue the following command from UNIX:

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

   The script sends the following files to the mainframe from the *DBSRVR_ROOT*\db2390\storproc\zos directory on Windows or the *$DBSRVR_ROOT*/db2390/storproc/zos directory on UNIX:

   - SIEBEL.UDF.RECEIVE.JCL
   - SIEBEL.UDF.LOAD.XMIT
   - SIEBEL.UDF.CNTL.XMIT
   - SIEBEL.UDF.DDL.XMIT
   - SIEBEL.UDF.DBRM.XMIT

3. Modify the *USER_ID*.SIEBEL.UDF.RECEIVE.JCL data set according to the instructions included in the JCL member, and submit it to receive the files in a readable format.

4. Before running the BIND, CREATE and WLMRFSH jobs, edit the UDF.DDL file and set the following values:

   - Set the Collid value in the exrate function to your table owner name
   - Set the WLM environment variable in the exrate and nextseq functions to the name of your application environment.

5. Edit the WLMRFSH file by entering an asterisk to the left of the TYPRUN parameter to comment-out this command.

6. Run the jobs in the UDF.CNTL PDS in the following order:

   - IEBCOPY
   - BIND
   - CREATUDF
   - WLMRFSH

   The WLMRFSH (refresh) job must be the last job run.

   Make sure that the return code equals 0.

7. Return to the midtier to continue the Siebel Schema installation by populating the schema with seed data.

Preparing for a Custom Installation

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

When you select the custom installation method, you must first create the storage control file that is used as the basis of your Siebel Schema custom installation. You can choose from the following options:

- Use an existing storage control file in the custom installation.

  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 and use it in the custom installation.

  You can edit the storage control file template directly, or you can use the Siebel Database Storage Configurator (dbconf.xls) to modify the template. For more information, see "Modifying Storage Control Files Using the Database Storage Configurator" on page 108.

- Extract a storage control file based on the layout of the existing Siebel Schema using the Extract Storage File option of the Database Server Configuration Utility.

  For more information, see "Extracting a Storage Control File from the DB2 Catalog."

Extracting a Storage Control File from the DB2 Catalog

You can create a storage control file to use during a custom installation using the Siebel extract utility. This utility extracts the storage layout of your source database from the DB2 catalog and creates a new storage control file that you can use 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. See "Validating the Extracted Storage Control File" on page 106.

Follow the instructions below to extract a storage control file under Windows and under UNIX respectively.

To extract the storage control file under Windows

1. Launch the Database Server Configuration utility and follow the steps in "Performing a Standard Installation Under Windows" until the Siebel Database Operation screen is displayed (Step 6 on page 86).
2. From the Siebel Database Operation screen, select Run Database Utilities, and then click Next.
3. From the Database Utility Selection screen, select Configure Database, and then click Next.
4. The following options are available on the Database Configuration Options screen:

   - Extract Storage File.

     Select this option to extract a storage control file based on the existing Siebel Schema layout.

   - Validate Storage File.

     Select this option to validate that a specified template and a specified schema will work on the DB2 host.

   Select Extract Storage File, and then click Next.

5. On the ODBC Data Source Name screen, you can either select 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 specify a different data source name.

   NOTE: The Siebel Server installation process automatically creates the data source, using the format SiebSrvr_*EnterpriseServerName*. To determine the name of your ODBC data source, navigate to Start > Settings > Control Panel > Administrative Tools > Data Sources (ODBC). Click the System DSN tab to view your ODBC data source name.

6. On the Database User Name screen, type your source database username and password, confirm the password by typing it again, and then click Next.

7. On the Siebel Schema Qualifier screen, type the schema qualifier 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.

8. On 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. The output is a new storage layout.

9. On the Default Database Name screen, enter the Default Database Name and the Default Table Space Name, and click Next.

10. On the Default Storage Group for Table Spaces screen, enter values to indicate the Default Storage Group for Table Spaces and the Default Storage Group for Indexes, and click Next.

11. On the Default Buffer Pool Name screen, specify the Default Buffer Pool Name for your table spaces and the Default Index Buffer Pool Name. Your DBA should have previously activated the buffer pools and granted access to them.

12. On the Database Encoding screen, indicate whether your DB2 subsystem is ASCII or EBCDIC.

13. In the Storage Control File screen, type the name you want to assign to the new storage control file, for example *DBSRVR_ROOT*\db2390*my_storage_file.ctl*, and then click Next.

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

15. Apply the configuration, or indicate that you want to apply it later:

    - To apply the configuration now, in the runnow message box, click OK.

    - To apply the configuration later, click Cancel. You can re-start the configuration later by navigating to the *SIEBSRVR_ROOT*\bin directory and entering the following from the command line:

      ```
      siebupg.exe /m master_dbconfig_extract_catalog_mf.ucf
      ```

16. In the Configuration Parameter Review screen, review the parameters that you entered, make any corrections necessary, and then click Finish.

17. When the Siebel Upgrade Wizard launches, click OK.

    The Siebel Upgrade Wizard creates the storage control file.

To extract the storage control file under UNIX

1. Launch the Database Server Configuration utility and follow the steps in "Performing a Standard Installation Under UNIX" until the Siebel Database Operation screen is displayed (Step 9 on page 92).
2. From the Siebel Database Operation screen, choose the 5 - Run Database Utilities option.
3. From the Database Utility Selection screen, choose the 3 - Configure Database option.
4. From the Database Configuration Options screen, choose the 1 - Extract Storage Control File option.
5. Enter the number that corresponds to the language in which you are running your database.
6. Enter your source ODBC connection name.
7. Enter your source database username and password.
8. In the Siebel Schema Qualifier screen, enter the schema qualifier in uppercase letters for your existing database or press ENTER to accept the default.

   The schema qualifier is also an authorization ID. The schema qualifier must start with a letter and cannot contain special characters.
9. In the Extract Options screen, select the Extract from Catalog option.
10. Enter the default Database Name or press ENTER to accept the default.
11. Enter the Default Table Space Name or press ENTER to accept the default.
12. On the Table Storage Group screen, enter values to indicate the default storage group for table spaces or press ENTER to accept the default.
13. On the Index Storage Group screen, specify the default storage group for indexes or press ENTER to accept the default.
14. On the Default Buffer Pool Name screen, specify the Default Buffer Pool Name for your table spaces.
15. On the Index Buffer Pool Name screen, specify the Index Buffer Pool Name. The buffer pools should already be activated and the DBA should have granted access to them.
16. On the Database Encoding screen, indicate whether your DB2 subsystem is ASCII or EBCDIC.
17. On the Storage File screen, type the directory path and name of the new storage control file, for example *DBSRVR_ROOT*/db2390/*my_storage_file.ctl.*
18. On the Common Parameters screen, indicate where you want to generate the log files. The default is dbconfig_extract_catalog_mf.
19. The configuration parameters you entered are listed for you to review. When you have verified them, enter 1 (Yes) to run the Upgrade Wizard (srvrupgwiz).

    The Siebel Upgrade Wizard creates the storage control file.

Validating the Extracted Storage Control File

You must validate the storage control file after you extract it or after you modify it. The validation produces two output files in the log directory under *SIEBSRVR_ROOT*:

- db_config_siebel.log
- db_config_validation.log

NOTE: When validating the extracted storage control file, the validation job calls for the schema.ddl file, as an example. 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. Launch the Database Server Configuration utility and follow the steps in "Performing a Standard Installation Under Windows" until the Siebel Database Operation screen is displayed (Step 6 on page 86).
2. On the Siebel Database Operation screen, select Run Database Utilities, and click Next.
3. Choose Configure Database, and click Next.

   The Database Configuration Options screen appears with the following options:

   - Extract Storage Control File
   - Validate Storage Control File
4. Choose the Validate Storage Control File option, and click Next.
5. On the ODBC Data Source screen, accept the default name of the ODBC data source (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 SiebSrvr_*EnterpriseServerName*. 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 to find the name of your ODBC data source.

   To continue, click Next.
6. On the Database User Name screen, indicate the source database user name and password. The Database User Name is an ID authorized to log in to the Siebel database. This user ID should have authorization to set CURRENT SQLID.

   To continue, click Next.
7. On the Siebel Schema Qualifier screen, type the identifier that designates the Siebel Schema for your database. This is also an authorization ID. The schema qualifier must start with a letter, can be up to 8 characters in length, and cannot contain special characters.

   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 preconfigured Schema, use ddl.ctl.

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

To continue, click Next.

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

10 In the runnow message box, click OK to run the configuration now. To run the configuration later, navigate to the *SIEBSRVR_ROOT*/bin directory and enter the following from the command line:

```
siebupg.exe /m master_dbconfig_validate_mf.ucf
```

11 On the Configuration Parameter Review screen, review the configuration values you entered on the previous Configuration Utility screens, make any changes necessary, and click Finish.

12 When the Siebel Upgrade Wizard launches, click OK to apply the configuration.

13 To check that the validated storage control file works, connect to the catalog, and run ddlimp in validation mode.

14 Review the log file (ddlimp.log) that is generated in the log subdirectory under *SIEBSRVR_ROOT*.

To validate the storage control file under UNIX

1 Launch the Database Server Configuration utility and follow the steps in "Performing a Standard Installation Under UNIX" until the Siebel Database Operation screen is displayed (Step 9 on page 92).

2 On the Siebel Database Operation screen, choose option 5 - Run Database Utilities.

3 On the Database Utility Selection screen, choose option 3 - Configure Database.

4 On the Database Configuration Options screen, choose option 2 - Validate Storage Control File.

5 Enter the number that corresponds to the language you are using.

6 Enter your ODBC data source name.

NOTE: This value 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.

7 Enter your source database username and password.

8 Enter the schema qualifier for your existing database (enter this value in all uppercase characters).

9 Enter the directory path of the schema file and then the name of the storage control file that you want to validate.

10 On the Common Parameters screen, specify the directory where you want the log files to be generated; the default is dbconfig_validate_mf.

11 Review the variables you entered, then enter 1 (Yes) to run the Upgrade Wizard (srvrupgwiz).

The storage control file is validated.

12 Review the log files generated by ddlimp (ddlimp.log) in the log directory under *SIEBSRVR_ROOT*.

Modifying Storage Control Files Using the Database Storage Configurator

When you install Siebel Business Applications, a Siebel Microsoft Excel tool, the Siebel Database Storage Configurator (dbconf.xls), is also installed. This tool helps you to configure your Siebel Schema by allowing you to import and edit the attributes of the Siebel-provided storage control file templates. You can also use this tool to configure an existing storage control file.

The Siebel Database Storage Configurator file is dbconf.xls; this file is installed in the *DBSRVR_ROOT*\db2390 (Windows) or $*DBSRVR_ROOT*/db2390 (UNIX) directory. To open dbconf.xls, you must have Microsoft Excel installed on your workstation.

After you edit a storage control file template using the Database Storage Configurator, save the file with a new name, and then direct the Database Server Configuration utility to your newly configured file when prompted to specify a storage control file during installation (see "Process of Performing a Custom Installation" on page 113). The output DDL is generated from the information in the (configured) storage control file that you specify.

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.

If you are not prompted to Enable Macros, verify that your Microsoft Excel security setting is Medium, as follows:

a Launch Excel, and choose Tools > Macro > Security.

b Set the security level to medium, and then restart your computer.

When the dbconf.xls 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 completed, 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.

   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.

6. Review the values, and update them as required.

   The parameters in the spreadsheet correspond to native DB2 syntax. For example, the parameters `PriQty` (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 to the object for which the tab exists. Thus, if no locksize is shown for a table space under the Tablespace tab, then that table space takes the default locksize from the database object it belongs to. For example, locksize at 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 Partitions column greater than zero, and then click Show TSPARTs.

     This displays the rows of corresponding partitions.

9 Edit the values as desired, and then move 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.

13 From the Home tab, click Clean, to make the Configurator ready for use with a different template.

A text box appears letting you know that you have cleaned all data successfully.

Validating Your Work

You can use the Database Storage Configurator Validation button at any time to validate the *syntax* in the dbconf.xls spreadsheet. Wait to validate until after you have finished editing to ensure your worksheets are consistent.

NOTE: The spreadsheet validation option is not a substitute for using the Database Server Configuration wizard to validate the storage control file after you amend it. The Validate Storage Control File option of the Database Server Configuration utility validates the new storage control file against the physical layout of the database.

To validate your dbconf.xls worksheet

1 After updating values for any of the template defaults, click the Validate button.

The Database 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.

NOTE: If you cannot view color indicators, refer to the *Accessibility Guide for Siebel Business Applications*.

2 Update any incorrect value, move off the cell, and revalidate.

The red highlighting disappears when the validation shows that the value is correct.

Partitioning Tables and Indexes Using the Database Storage Configurator

You can 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 dbconf.xls spreadsheet validation option is not a substitute for using the Database Server Configuration wizard to validate the storage control file after you amend it by partitioning tables. The Validate Storage Control File option of the Database Server Configuration utility validates the new storage control file against the physical layout of the database.

Decreasing the Number of Partitions

This topic describes how to decrease the number of partitions in a table space.

To decrease the number of partitions

1. Launch the Database Storage Configurator, and import the .ctl file you want to amend as described in "Modifying Storage Control Files Using the Database Storage Configurator" on page 108.
2. Click the Tablespace tab on the Database Storage Configurator.
3. Locate the appropriate Tablespace, and change the value in the Partitions column to reflect the number of partitions you want to create.
4. Save the new value by stepping off the cell.
5. Click Show TSPARTs.

   If a message box appears indicating that there are additional table space partitions for this table, and requesting that you review and delete them, do so now.

   Otherwise, 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.

   Otherwise, 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 only enter a single quote, Excel does not save it.
10. Make any other edits needed to the template you are using and follow the steps under "Modifying Storage Control Files Using the Database Storage Configurator" on page 108 to verify and save your changes.

Increasing the Number of Partitions

Usually, partitioning is carried out at the table level. However, you must also adjust the related table space and partitioning index. The following procedure describes how to increase the number of partitions for a table and an index in a given table space.

To increase the number of partitions

1. Launch the Database Storage Configurator as described in "Modifying Storage Control Files Using the Database Storage Configurator" on page 108, 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 locate 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 the new value, 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 amending.

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 "Modifying Storage Control Files Using the Database Storage Configurator" on page 108 to verify and save your changes.

Working with CLOBs Using the Database Storage Configurator

Your Siebel application is delivered with the objects needed to create character large objects (CLOBs). You can use these objects to create CLOBs at the table level only, to avoid excessive processing overhead.

To activate a CLOB

1 Launch the Database Storage Configurator as described in "Modifying Storage Control Files Using the Database Storage Configurator" on page 108 and select the Table tab.

2 Review the contents of the CLOB column.

3 To activate a specific CLOB, type Yes in the appropriate row in the CLOB column, and then save the value by stepping off the cell.

NOTE: The Auxiliary Tablespace tab 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 you can change for the auxiliary object.

4 Make any other changes needed to the template you are using. Then follow the steps under "Modifying Storage Control Files Using the Database Storage Configurator" on page 108 to verify and save your changes.

Process of Performing a Custom Installation

You can use the Database Server Configuration utility option, Customized Install, to configure the storage layout for the Siebel application installation.

To perform a custom installation, perform the following tasks:

- Prepare a storage control file. (See "Preparing for a Custom Installation" on page 102.)
- Run the Database Server Configuration utility, select the Customized Install option, and specify that the utility uses your customized storage control file in the storage configuration for the Siebel Schema. (See the following procedure for your operating system.)
- Import the Siebel Repository. (See Chapter 9, "Siebel Repository Import and Postinstallation Tasks.")

Performing a Custom Installation Under Windows

Perform the following procedure if you are installing Siebel Business Applications on the Windows operating system.

To perform a custom installation under Windows

1 Launch the Database Server Configuration utility and follow the steps described in "Performing a Standard Installation Under Windows" on page 85 until the Select Installation Type screen appears (Step 12 on page 87).

2 Select the Customized Install option and then click Next.

3 In the Storage Control File screen, use the Browse button to locate the customized storage control file that you want to use to configure storage on the DB2 host.

To continue, click Next.

4 In the Select Installation Mechanism screen, indicate which installation mechanism you want to use:

- Generate DDL into Files

  Select this option to generate the DDL required to create the Siebel Schema into 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 5.

- Run DDL Automatically

  Select this mechanism 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 method.

  To continue, click Next and proceed to Step 7.

5. On the DDL Commit Frequency screen, choose the number of DDL statements that are issued between issuing COMMITs, and click Next to continue.
6. In the Output Directory screen, either accept the default directory, `C:\sea78\dbsrvr\db2390\dboutput\install`,or enter a different directory to indicate where you want to save the generated DDL for later transport to the DB2 host.
7. On the Log Output Directory screen, indicate the directory where you want to generate the log files. Either accept the default value, `install_mf`, or enter a new directory name.
8. In the runnow message box, choose one of the following options:
   - To apply the configuration now, click OK.
   - To apply the configuration later, click Cancel.

     To apply the configuration later, from the command line, navigate to the *SIEBSRVR_ROOT*\bin directory and enter the following command:

     ```
     siebupg.exe /m master_install_mf.ucf
     ```

     NOTE: This command is generated dynamically and varies, depending on the location of your Siebel Server installation.
9. In the Configuration Parameter Review screen, review the configuration values you entered and then either select Previous to return to previous screens so that you can amend the parameter values, or select Finish to accept the existing values.
10. When the configuration has completed successfully, click OK.
    - This step completes the installation option to Run DDL Automatically.
    - If you chose the Generate DDL Into Files installation option in Step 4 on page 113, you must now apply the Siebel Schema on the DB2 host, validate the schema, install user-defined functions, and populate the Siebel Schema with seed data.

      For information on performing these tasks, see the section, To Complete the Siebel Schema Installation Using Generated DDL, in "Performing a Standard Installation Under Windows" on page 85.

When you have successfully completed the installation, you are ready to import the Siebel Repository. See "Importing the Siebel Repository" on page 119 for further information.

About Reviewing the Log Files for Windows

The Siebel Database Server installation and configuration process creates a number of log files, such as upgwiz.log and upgwiz_01.log (the numeric suffix incrementing for subsequent log files), within the *SIEBSRVR_ROOT**log* directory. Review the log files for any errors.

When you install your base language pack, it creates a log file named dataimp_prim_lang.log.

The main log file, produced by the Siebel Upgrade Wizard (upgwiz.log), does not contain information on errors returned by applications. You can find additional information about the configuration process in the sw_cfg_util.log file. Depending on the operation you perform, the ddl_ctl.log file reflects any errors.

To find additional operation-specific log files, open upgwiz.log using a text editor and search for a ".log" string. Upgwiz.log shows the names of log files produced during the installation operation; for example, if the configuration of the Database Server fails, the upgwiz.log file contains the following information:

```
GenericLog GenericError 1 2002-02-28 15:51:24 (err=1) was returned by application
(C:\sea78\siebsrvr\bin\ddlimp.exe ...)

GenericLog GenericError 1 2002-02-28 15:51:24 Execute file action failed
(err=Launching the application)
```

In an example such as the previous one, the ddlimp.log file contains a detailed error message:

```
DDLIMP-ERR-1071: Unable to build ddl statement "add col" (Open unloadtbl.jcl)
```

About the htmlgenm Utility

You can also use the Siebel utility, htmlgenm, to create an HTML file with a summary report of all steps performed by the Upgrade Wizard, together with information on log files.

The htmlgenm utility has three options:

/M Master UCF file	Default: master.ucf
/H Summary HTML file	Default: summary.htm
/L Siebupg.exe LOG file	Default: upgwiz.log

For example, to view the HTML results from the database installation, use the following command to generate the summary.htm file in the log subdirectory:

```
htmlgenm /m master_install_mf.ucf
```

Acceptable Errors and the errors.rtf File

The Siebel Database Server installation and configuration 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. The errors.rtf file is located in the *DBSRVR_ROOT*\db2390 directory.

- If a log file is not listed in the errors.rtf file, then there are no acceptable error messages for that log file.
- If the log files contain errors listed in the errors.rtf file, no further action is required.

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.

When reviewing error messages, be aware that error numbers can change after the installation of a new driver version. Compare the error descriptions to find out which are acceptable errors for the z/OS platform.

CAUTION: Although you are unlikely to encounter errors other than those listed in the errors.rtf file, it is critical that you review the error messages. Certain errors, such as a failure to create indexes, can result in performance problems or anomalous behavior in Siebel Business Applications.

Performing a Custom Installation Under UNIX

Perform the following procedure if you are installing Siebel Business Applications on the UNIX operating system.

To perform a custom installation under UNIX

1. Launch the Database Server Configuration utility and follow the steps described in "Performing a Standard Installation Under UNIX" on page 91 until the Select Installation Type screen appears (Step 16 on page 93).
2. Select option 2 - Customized Install.
3. In the Storage Control File screen, enter the name of the customized storage control file that you created and saved earlier, including the full path to its location. This file is used to configure storage on DB2.
4. Specify the mechanism you want to use for installation:
   - 1 - Generate DDL Into Files.

     Generates 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 (schema.sql and ddlview.sql) to the DB2 host where the DBA executes it, using customary methods, for example, SPUFI or DSNTEP2. Proceed to Step 5 on page 117.

- 2 - Run DDL automatically.

  Select this option 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 7.

5. On the DDL Commit Frequency screen, choose the number of DDL statements to be processed before a COMMIT statement is issued.
6. Enter the path to the output directory for the DDL files that you will transfer to your z/OS machine, or accept the default directory, `dbsrvr/db2390/dboutput/install`.
7. Indicate the directory where you want the log files to be generated. Either accept the default value, `install_mf`, or enter a new directory name.
8. Review the configuration values you entered for accuracy. If the values are correct, launch the Siebel Upgrade wizard (srvrupgwiz) by entering 1 (Yes).
   - This step completes the installation option to Run DDL Automatically.
   - If you chose the Generate DDL Into Files installation option in Step 4 on page 116, you must now apply the Siebel Schema on the DB2 host, validate the schema, install user-defined functions, and populate the Siebel Schema with seed data.

     For information on performing these tasks, see the section, To Complete the Siebel Schema Installation Using Generated DDL, in "Performing a Standard Installation Under UNIX" on page 91.

When you have successfully completed the custom installation, you are ready to import the Siebel Repository. See "Importing the Siebel Repository" on page 119 for further information.

About Reviewing the Log Files for UNIX

The Siebel Database Server installation and configuration creates a number of log files, such as srvrupgwiz.log and srvrupgwiz1.log (which increments to additional logs) within the *$SIEBSRVR_ROOT/log* subdirectory, which you can review for any errors.

When you install your base language pack, it creates a special log file, 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_util.log file. Depending on the operation performed, the 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 the 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
(/sea78/siebsrvr/bin/ddlimp.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, such as the following:

```
DDLIMP-ERR-1071: Unable to build ddl statement "add col" (Open unloadtbl.jcl)
```

For further information on reviewing Siebel Schema installation and configuration error messages, see “Acceptable Errors and the errors.rtf File” on page 116.

Rerunning the Installation

If you need to rerun the Siebel Schema installation from the beginning, either after completing an installation run or after starting but not finishing the run, enter a new log output directory name.

You have to rerun the installation when you generate new DDL for the Siebel Schema based on a new or modified storage control file.

9 Siebel Repository Import and Postinstallation Tasks

This chapter describes how to import the Siebel Repository and describes other mandatory and optional postinstallation procedures that you perform to complete the installation process.The chapter consists of the following topics:

- "Process of Completing the Database Server Installation" on page 119
- "Importing the Siebel Repository" on page 119
- "Postinstallation Tasks for the Siebel Repository" on page 124

Process of Completing the Database Server Installation

When you have installed the Siebel Database Server files and the Siebel Schema, you must perform the following tasks to complete the Database Server installation and configuration process:

1. "Importing the Siebel Repository" on page 119
2. "About Reviewing the Log Files for Repository Import" on page 124
3. "Granting Table Privileges" on page 125
4. "Loading Statistics" on page 125
5. "Validating the Siebel Schema" on page 125
6. (Optional) "Installing Multilingual Seed Data" on page 127
7. (Optional) "Importing a New Language to Your Repository" on page 130
8. "Populating the Siebel File System" on page 132

Importing the Siebel Repository

Importing the Siebel Repository 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 Business Applications, Release 7.8 application objects.

Regardless of how many Siebel Business Applications you are using (for example, Siebel Sales, Siebel Service, Siebel Marketing), you load data into the repository tables only once per installation.

You can also export the Siebel Repository 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 and follow the steps in "Performing a Standard Installation Under Windows" until the Siebel Database Operation screen is displayed (Step 6 on page 86).

2 Select the Import/Export Repository option, and click Next.

3 On the Select Repository Operation screen, select one of the following options, as appropriate:

- Import Repository

  To import the Siebel Repository for the first time with a base language, select this option. Proceed to Step 4.

- Add Language to an Existing Repository

  If you have 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 130.

  NOTE: You must have 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. Proceed to Step 6.

  NOTE: You can also use the Repository export option to replicate repositories.

To continue, click Next.

4 On the Import Selection screen, indicate whether you want to import a standard Siebel 7 repository, or a customized repository, by clicking the appropriate 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.

5 On the Language Selection screen, select the language in which you want to run the database.

6 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 to the one you currently point to, you can enter the ODBC name of that database server, provided it is registered 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.

7 In the Database User Name screen, enter the database user name and the associated password.

The Database User Name is 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.

To continue, click Next.

8 On the Siebel Schema Qualifier screen, enter the ID that identifies the Siebel Schema owner, for example, SIEBTO, and click Next.

- If you are importing your repository for the first time, the Import Repository Name screen appears. Proceed to Step 9.
- If you are exporting your repository, the Export Repository Name screen appears. Proceed to Step 10.

9 On the Import Repository Name screen, indicate the following:

- Import Repository Name

  Enter the designated name for this Siebel Repository.

- Repository File Name

  Accept the default installation path and file name for this repository or type another valid installation path.

To continue, click Next. Proceed to Step 11.

10 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.

11 On the Log Output Directory screen, accept the default directory name (imprep) or type another valid name.

12 When the runnow message box appears, click OK to apply the configuration now, or click Cancel to apply it later.

To rerun the Siebel Upgrade Wizard later, navigate to the *SIEBSRVR_ROOT*\bin directory and enter the following from the command line:

```
siebupg.exe /m master_imprep.ucf
```

13 On the Configuration Parameter Review screen, review the configuration values that you entered on the previous configuration screens, and then click Finish.

The Upgrade Wizard launches.

14 Click OK to begin the repository import/export or Cancel to exit.

A window appears, displaying information about the repository import (export) activities. Afterwards, a message appears when the repository import (export) is completed.

To verify that the import was successful, review the log files. See "About Reviewing the Log Files for Repository Import" on page 124.

TIP: Locate the file master_imprep.ucf in the *SIEBSRVR_ROOT*\ bin directory and print it out, so that you have a record of the values you entered.

To import the Siebel Repository under UNIX

1 Launch the Database Server Configuration wizard and follow the steps in "Performing a Standard Installation Under UNIX" until the Siebel Database Operation screen is displayed (Step 10 on page 92).

2 Select the Import/Export Repository option.

3 On the Repository Operations screen, choose one of the following options, according to the procedure you want to complete:

- Import Repository

  To import the Siebel Repository for the first time with a base language, select this option. Proceed to Step 4.

- Add Language to an Existing Repository

  If you have 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 130.

  NOTE: You must have 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 necessary. Proceed to Step 6 on page 123.

  NOTE: Repository export can also be used to replicate repositories.

4 If you select the Import Repository option, the Import Selection screen appears. Specify whether you want to import a standard Siebel 7 repository, or a customized repository, by entering the number associated with the option.

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.

Proceed to Step 5.

5 Select the language in which you want to run the database. (This is your base language.)

6 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 (you must have registered this alias within DB2 Connect).

NOTE: If you want to work with information on a different database server to the one you currently point to, you can enter the ODBC name of that database server, provided you have 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*.

7 Specify the database user name and password.

This is the ID that is used to log into the Siebel Database Server. This user ID should have authorization to set the CURRENT SQLID, for example, SADMIN.

8 Specify the Siebel Schema Qualifier, for example, SIEBTO.

This value is the identifier that designates the Siebel Schema and is also an authorization ID. The Schema Qualifier must be in uppercase, must start with a letter, cannot contain special characters, and can be up to eight characters in length.

- If you are importing your repository for the first time, the Import Repository Name screen appears. Proceed to Step 9.
- If you are exporting your repository, the Export Repository Name screen appears. Proceed to Step 11.

9 On the Import Repository Name screen, indicate the designated name for this Siebel Repository.

10 Enter the default installation path and file name for the repository or type another valid installation path.

Proceed to Step 12.

11 In the Export Repository Name screen, enter 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.

12 On the Common Parameters screen, accept the default directory name (imprep) or type another valid name to indicate where log output files are to be created.

13 Review the configuration values that you entered on the previous configuration screens and then enter 1 (Yes) to launch the Siebel Upgrade wizard (srvrupgwiz).

If a program or system error occurs and you have 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 the following from the command line:

```
    srvrupgwiz /m master_imprep.ucf
```

If the repository import (export) is successful, the Upgrade Wizard displays a message that the repository has been imported (or exported). Otherwise, see the log files.

About Reviewing the Log Files for Repository Import

The repository import process creates a number of log files within the *SIEBEL_ROOT*\log directory (Windows) or the $*SIEBEL_ROOT*/log directory (UNIX) that you can review for any errors that occur. The log file names for subsequent processes include a numeric suffix that increments for additional log 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 if the Upgrade Wizard encounters a problem and you attempt to import (export) the repository again.

When you import a repository with a new language, it creates the following log files:

- imprep_prim.log
- imprep_lang.log

For more details on logs, refer to the discussion under "About Reviewing the Log Files for Windows" on page 115, "Acceptable Errors and the errors.rtf File" on page 116, and "About Reviewing the Log Files for UNIX" on page 117.

Troubleshooting the Repository Import

If the Siebel Repository import fails midway through the process, you must clean up the data from the failed import using Siebel Tools to delete the repository.

Postinstallation Tasks for the Siebel Repository

After importing the Siebel repository, you still have a number of tasks to perform before your Siebel Business Applications implementation on DB2 for z/OS is completed. The following are the mandatory tasks you must perform:

- "Granting Table Privileges" on page 125
- "Loading Statistics" on page 125
- "Validating the Siebel Schema" on page 125
- "Populating the Siebel File System" on page 132

You might also have to perform the following tasks, depending on your implementation:

- "Installing Multilingual Seed Data" on page 127
- "Importing a New Language to Your Repository" on page 130

Granting Table Privileges

The database administrator edits and executes the file, grantstat.sql, located within the *DBSRVR_ROOT*\db2390 directory (Windows) or the *$DBSRVR_ROOT/*db2390 directory (UNIX), 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 another user ID.
3. Execute the grantstat.sql script using the method you prefer.

NOTE: If you use multiple security groups, grant this privilege to each group.

Loading Statistics

Execute the script, loadstat.sql, to load the statistics required for the 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 *DBSRVR_ROOT*\db2390 directory (Windows) or the *$DBSRVR_ROOT/*db2390 directory (UNIX).
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 prefer.

Validating the Siebel Schema

After you install the Siebel Database Server and import the repository, you must compare the physical database schema with the repository to make sure that there are no inconsistencies between them. Use the Siebel Server utility, dbchck.exe (Windows) or dbchck (UNIX), located in the bin subdirectory of your Siebel Server installation directory, to make this comparison.

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 `/dict /all`. 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 *SIEBSRVR_ROOT*\bin directory (Windows) or the *$SIEBSRVR_ROOT/*bin directory (UNIX), locate dbchck.exe (Windows) or dbchck (UNIX).
2. (Windows only) Delete the dictionary cache file (diccache.dat) before running dbchck, and verify that there are no EIM or Siebel Remote operations running.

   The dbchck utility creates a new diccache.dat file before carrying out the integrity check. By deleting the existing diccache.dat file before starting dbchck, you ensure that dbchck validates against the Siebel Repository you specify.

   NOTE: If you stop any EIM or Siebel Remote processes, you can restart them after dbchck has run.

3. Source environment variables as appropriate for your platform, using either siebenv.bat (Windows) or siebenv.sh or siebenv.csh (for UNIX).

   Environment variable scripts are located in the *SIEBSRVR_ROOT*\bin directory (Windows) or the *$SIEBSRVR_ROOT/*bin directory (UNIX).

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 /dict /all /s ODBC data source
   ```

   where:

   - *password* = login password to the database.
   - *Your Siebel Repository Name* = the repository 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.
   - *SIEBTO* = the Siebel Schema Qualifier.

   NOTE: To view all the dbchck parameters with their descriptions, use option /h.

   Any discrepancies found appear on the screen. Detailed information is written to a log file in the *siebsrvr*\bin directory (Windows) or the $*siebsrvr/* bin directory (UNIX).

   NOTE: The log file name is the name you specified after the log file parameter (/l). In the previous example, the file is named dbchck.log.

6. Review the log file generated as a result of running this script. Any discrepancies are flagged as failures. Investigate all discrepancies.

Discrepancies in the dbchck Log

After completion of the dbchck.exe script, the dbchck.log file generated may contain discrepancies noted as failures. However, some failures are expected due to the index length restriction, such as the following:

```
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)
```

This message is caused by the index length restriction of 255 bytes that applies to indexes created on DB2 for z/OS v7. Whether or not the truncated indexes cause an issue depends on your implementation and business requirements. Review the truncated indexes to determine whether these need to be addressed.

Some errors noted in the dbchck.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: REVISED_COST  numeric(22,7) null
physical:   REVISED_COST  decimal(10,0) null
```

This error is caused by a mismatch in the data type definition of the column in the logical and physical schema. The definition of a column in the logical and physical schema should be synchronized; you can synchronize the logical definition with the physical schema using Siebel Tools. After the synchronization is done, run dbchck.exe again.

For more information about using the dbchck utility, 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). You install this seed data by adding new language packs to your database after you install the base language for your Database Server. These language packs populate the List of Values (LOV) with seed data in the new language.

You must successfully install seed data in your base language before you can add seed data in other languages to your database.

NOTE: 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 for "Installing the Database Server Software Under Windows", specifying the installation instance to which you want to add the language (Step 5 on page 73).

2 When installation is completed, launch the Database Server Configuration utility by following the instructions under "Performing a Standard Installation Under Windows" on page 85.

3 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.

4 In the Base Language screen, specify which of the installed languages is the one in which you want to primarily run your database. (This is 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.

5 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.

6 Respond to the remaining prompts as you did when you originally completed the procedure, "Performing a Standard Installation Under Windows" on page 85, until you have completed Step 11 on page 87. The Repository Name screen appears.

7 Type the name of your Siebel Repository or accept the displayed default name.

   To continue, click Next.

8 On the Log Output Directory screen, accept the default name, or type another valid name to indicate where you want log output files to be created.

9 When the runnow message box appears, click OK to apply the configuration now, or click Cancel to apply it later.

   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_ROOT*\bin and entering:

```
siebupg.exe /m master_install_lang.ucf
```

10 In the Configuration Parameter Review screen, review the configuration values you entered and then click Finish. The items to be executed or imported are displayed.

11 To begin, click OK.

A window appears, displaying information about installation activities. Afterwards, a message appears when the installation is completed.

To verify that the installation was successful, review the log files. See "About Reviewing the Log Files for Repository Import" on page 124.

TIP: Locate the file master_install_lang.ucf in the *SIEBSRVR_ROOT*\bin directory and print it out, so that you have a record of the values you entered.

12 When you have finished configuration using the Database Server Configuration wizard, you must enable the multilingual list of values (MLOV) capability within Siebel Business Applications. For information on how to do this, see the *Global Deployment Guide*.

13 After you enable MLOV capability, proceed to "Importing a New Language to Your Repository" on page 130.

To install multilingual seed data under UNIX

1 Follow the instructions under "Installing the Database Server Software Under UNIX", specifying the installation instance to which you want to add the new language (Step 7 on page 76).

2 Select the language you want to add to the current installation and click Next. The new language is installed.

3 When installation is successfully completed, click Finish.

4 Launch the Database Server Configuration utility by following the instructions under "Performing a Standard Installation Under UNIX" on page 91.

5 When the Installation Options screen is displayed (Step 9 on page 92), select option 2 - Add a language to an existing Siebel Database.

6 On the Language Selection screen, specify which of the displayed languages you are adding on top of your base language for this database.

7 On 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 your base language.)

8 Respond to the remaining prompts as before until you have completed Step 14 on page 93. The Repository Name screen then appears.

9 Either accept the default repository name or enter another name.

10 On the Common Parameters screen, indicate where log output files are to be created. The default directory is install_lang.

11 Review the values you previously entered for accuracy. If they are correct, launch the Siebel Upgrade Wizard (srvrupgwiz) by entering 1 (Yes).

A message appears when the installation is completed.

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 *$SIEBSRVR_ROOT/*bin and entering:

```
srvrupgwiz /m master_install_lang.ucf
```

To verify that the installation was successful, review the log files. See "About Reviewing the Log Files for Repository Import" on page 124.

TIP: Locate the file master_install_lang.ucf in the $*SIEBSRVR_ROOT*/bin directory and print it out, so that you have a record of the values you entered.

12 You must now enable the multilingual list of values (MLOV) capability within Siebel Business Applications. For information on how to do this, see the *Global Deployment Guide*.

13 After you enable MLOV capability, 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 Business Applications to better operate in the new language.

Regardless of how many Siebel Business 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 to your database after you complete your initial installation with your initial set of languages, leave the Gateway Name Server Address and Enterprise Server Name fields blank when prompted for these values by the Database Server Configuration utility. If you enter values in these fields, an error occurs.

To import a repository in a secondary language under Windows

1 After installing multilingual seed data (see "Installing Multilingual Seed Data" on page 127) launch the Database Server Configuration utility.

2 Follow the steps in "Importing the Siebel Repository" on page 119 to Step 3 (Select Repository Operation) and then select the Add Language to an Existing Repository option.

   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 steps in listed in "Importing the Siebel Repository" on page 119 up to Step 9 on page 121.

5 In the Import Repository Name screen, indicate the following:

   - Import Repository Name

     Type the designated name for this Siebel Repository or accept the default.

- Localized Repository File Name

  If you are importing a secondary language repository on top of your base-language repository, either accept the default installation path and file name for this repository or type another valid installation path.

To continue, click Next.

6 On the Log Output Directory screen, accept the default name or type another valid name to indicate where log output files are to be created.

7 When the runnow message box appears, click OK to apply the configuration now, or click Cancel to apply it later.

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 the *SIEBSRVR_ROOT*\bin directory and entering:

```
siebupg.exe /m master_imprep_lang.ucf.
```

8 In the Configuration Parameter Review screen, review the configuration values you entered on the previous configuration screens and then click Finish.

The Database Server Configuration utility displays the item to be imported.

9 To begin, click OK.

A window appears, displaying information about installation activities. A message appears when the installation is completed.

To verify that the installation was successful, review the log files. See "About Reviewing the Log Files for Repository Import" on page 124.

TIP: Locate the file master_imprep_lang.ucf in the *SIEBSRVR_ROOT*\bin 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 in the UNIX section of "Importing the Siebel Repository" on page 119 up to Step 3 on page 122 (Repository Operations screen).

2 Select option 2 - Add language to an existing Repository.

3 In the Language Selection screen, specify the new repository 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 the procedure described in the UNIX section of "Importing the Siebel Repository" until the Import Repository Name screen is displayed (Step 9 on page 123).

5 Enter the name of the repository you want to import.

6 On the Repository File Name screen, if you are importing a secondary language repository on top of your base-language repository, specify the default installation path and file name for this repository or type another valid installation path.

7. On the Common Parameters screen, accept the default name or type another valid name to indicate where log output files are to be created.

8. Review the values you previously entered for accuracy. If the values are correct, launch the Siebel Upgrade Wizard (srvrupgwiz) by entering 1 (Yes).

   If a program or system error occurs you can rerun the Wizard, starting at the point at which the wizard failed, by navigating to $*SIEBSRVR_ROOT*/bin and entering:

   ```
   srvrupgwiz /m master_imprep_lang.ucf
   ```

   To verify that the installation was successful, review the log files. See "About Reviewing the Log Files for Repository Import" on page 124.

   TIP: Locate the file master_imprep_lang.ucf in the *$SIEBSRVR_ROOT*/bin directory and print it out, so that you have a record of the values you entered.

Populating the Siebel File System

Specific files required to run the Siebel File System, such as correspondence templates and Siebel Marketing files, are provided with the Siebel Database Server software. A directory called *DBSRVR_ROOT*\files (Windows) or *$DBSRVR_ROOT*/files (UNIX) is created automatically when you install the Siebel Database Server.

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 *DBSRVR_ROOT*\files (Windows) or *$DBSRVR_ROOT*/files (UNIX) directory to the att subdirectory of the Siebel File System.

2. Copy the KB.kb file from the *DBSRVR_ROOT*\files (Windows) or *$DBSRVR_ROOT*/files (UNIX) directory to the cms directory within the Siebel File System.

3. Verify that the files are where they should be.

10 Customizing a Development Environment

This chapter describes procedures for customizing development environments and migrating those customizations to user-acceptance or production environments on DB2 for z/OS.

This chapter consists of the following topics:

About Customizing Your Development Environment

Customization of Siebel Business Applications within your development environment can involve:

- 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 the *Using Siebel Tools* guide and the *Siebel Business Process Designer Administration Guide*. However, several customization procedures and issues are specific to DB2 for z/OS; these are described in this chapter.

It is assumed that you perform your development on either of the following:

- DB2 on a Windows, AIX, or Solaris machine
- DB2 for z/OS on a partition of your z/OS 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 for z/OS

There are certain considerations and procedures to follow in Siebel Tools when developing applications that run against databases on DB2 for z/OS. These include:

- "Siebel Tools Configuration File Parameters"
- "Setting Database Options"
- "Storage Control File Names" on page 135
- "About Inactivating Unused Indexes" on page 135
- "Truncation of Unique Index Columns" on page 135
- "About Reducing VARCHAR Field Lengths" on page 136
- "How Siebel Systems Implements LONG Columns on DB2 for z/OS" on page 136
- "About Using Character Large Objects (CLOBs)" on page 137
- "Changing LONGVARCHAR Data Type to VARCHAR Data Type" on page 139

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 for z/OS, as shown in Table 14.

Table 14. 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 for Windows) that will eventually be deployed on DB2 for z/OS, you must set the appropriate database options in Siebel Tools. Setting these options enables all Siebel Tools features for DB2 for z/OS and validates Siebel objects for the zSeries:

- The names of schema objects are limited to 18 characters (DB2 for z/OS v7 only).
- 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 for z/OS

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.

Storage Control File Names

When working in Siebel Tools, you must specify the complete paths for storage control file names in the Apply dialog box when applying schema changes to a database. (In previous releases, the storage control file was referred to as the table groupings file.)

About Inactivating Unused Indexes

The standard Siebel data model has a large number of indexes that can degrade performance on DB2 for z/OS. This performance degradation can occur 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 for z/OS 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.

Truncation of Unique Index Columns

Indexes on DB2 for z/OS v7 are limited to 255 characters (this limitation does not apply to DB2 for z/OS v8).

If you customize your development environment without setting the database options for DB2 for z/OS, your indexes might be truncated upon migration to the DB2 host and could become nonunique. For more information, see "Setting Database Options" on page 134.

When you analyze your business needs, if the unique indexes seem too long, it is recommended that you reduce the field lengths.

About Reducing VARCHAR Field Lengths

DB2 for z/OS pads VARCHAR fields to their maximum lengths in indexes. To increase flexibility, Siebel applications make extensive use of VARCHAR columns. This use of VARCHAR columns can cause performance degradation for z/OS 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 reduce the length of participating columns. To make better use of index space, users of DB2 for z/OS can reduce VARCHAR lengths, in cases where there is no application impact, using Siebel Tools.

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 the *Using Siebel Tools* guide.

How Siebel Systems Implements LONG Columns on DB2 for z/OS

Siebel Business Applications use LONG columns on Database Servers to store arbitrarily long character data up to 16350 characters. Examples include scripts, emails, notes, and descriptions. On all RDBMS platforms, except DB2 for z/OS, 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 for z/OS, a LONG column defined in the Siebel Schema is mapped to a LONGVARCHAR column, which always has a size of 32700 bytes.

On DB2 for z/OS, 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.

You can estimate the size of the LONGVARCHAR column by calculating the size of the table space minus the size of all non-LONGVARCHAR columns; the LONGVARCHAR column takes up the remainder of the space 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, results in the LONGVARCHAR column having a size of about 11 KB. For more information on calculating the size of LONGVARCHAR columns, refer to the vendor documentation on the IBM Web site.

How Siebel Systems Stores Tables with LONGVARCHAR Columns

If Siebel Systems created all LONGVARCHAR columns on DB2 for z/OS with a length of 16350 characters, this would effectively force all tables to a 32-KB table space; this table space size increases buffer pool storage requirements as it requires twice as much storage as a 16-KB table space. To minimize the number of tables created in 32-KB table spaces, during the Siebel installation process, tables with LONGVARCHAR columns are created in a 16-KB table space by default. Siebel Business Applications assume that LONGVARCHAR columns have a logical length of 4046 bytes, which forces the table to be created in a 16-KB table space.

NOTE: Although 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 for z/OS.

By taking this approach, almost all aspects of the Siebel application function without 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 following choices:

- Move the table to a 32-KB table space.

  If you move a table to a 32-KB table space, the LONGVARCHAR column can be up to 30 KB in length, doubling the storage requirements for buffer pools. Moreover, the result set of a join SELECT statement containing a LONGVARCHAR column can exceed 32 KB. When the result set does exceed 32 KB, you may run into the 32-KB in-memory sort limit of DB2 for z/OS v7. When DB2 for z/OS performs an in-memory sort, the full length of the LONGVARCHAR column is added to the total row width. As a result, you may experience SQL0670N errors.

- Use a CLOB column instead of a LONGVARCHAR column.

  For further information, see "About Using Character Large Objects (CLOBs)."

About Using Character Large Objects (CLOBs)

If you choose, you can store Siebel application objects defined as LONGVARCHARs as character large objects (CLOBs) on DB2 for z/OS. Siebel Business 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 for z/OS environment. However, if you have Siebel Marketing, change the LONGVARCHAR 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. However, CLOBs are turned off by default. You must turn them on for each table by setting `Clobs = Yes` in the storage control file.

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.

Issues in Using a CLOB Column Instead of a LONGVARCHAR Column

If you use a CLOB column instead of a LONGVARCHAR column, you do not run into the 32-KB in-memory sort limit of DB2 for z/OS v7. This memory limitation does not occur because DB2 for z/OS does not account for the actual length of the CLOB column in estimating the total row width to be sorted in memory. However, using a CLOB column has an impact on performance and storage (DASD). Fetching data from CLOB columns requires extra network flows between the client and Database Server, which can impact response time. Also, writing data to CLOB columns is slower than writing to LONGVARCHAR columns. So, for tables that are frequently used, moving to CLOB columns is not an option.

Using CLOB columns generally increases the amount of storage (DASD) needed, because space is allocated for the total width of the CLOB column even if the CLOB column contains only one byte of data.

IBM and third-party vendor tools have limited or no support for CLOB columns. Compression is also not supported for LOB table spaces.

Aside from performance and storage tradeoffs, perhaps the most important tradeoff for using CLOBs relates to 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. You need to consider this additional overhead before deciding to use CLOB columns.

For more information on the using LONGVARCHARs and CLOBs, see the *Upgrade Guide for DB2 UDB for z/OS*.

Siebel Tables Defined with LONG Columns in 32-KB Table Spaces

The following tables are defined in 32-KB table spaces in the Siebel application. Siebel Business Applications stores these tables in a 32-KB table space because the LONG columns in these tables need as much space as possible.

- S_BITMAP_DATA
- S_SCHMST_DBSCPT
- S_SERVICE_SCRPT
- S_NOTE
- S_SRC
- S_DMND_CRTN_PRG

Siebel Tables Defined with CLOB Columns

The following tables use CLOB columns in the Siebel application. Siebel Business Applications use CLOB columns for these tables because they belong to the Siebel repository and read and write activity to them is small. CLOB columns are also used if a table cannot be stored in a 32-KB table space because of the 32-KB in-memory sort limit of DB2 for z/OS v7.

- S_DMND_CRTN_PRG
- S_NOTE
- S_BITMAP_DATA

Changing LONGVARCHAR Data Type to VARCHAR Data Type

To avoid ambiguity in the actual length of the underlying field, you can change LONGVARCHAR data types to VARCHAR data types on the physical schema.

To change a LONGVARCHAR data type to a VARCHAR data type

- Set the data type of Data Public type tables and their columns of type LONGVARCHAR to VARCHAR(n) at the physical schema level.

CAUTION: Do not change the data type or width in the logical repository and apply the changes from Siebel Tools as errors can occur.

When performing schema synchronization, schema migration or upgrade tasks, ensure that rebuild statements are removed from the schema.sql file.

You do not need to modify the data type of Repository or System type tables.

Converting Nonpartitioned Tables to Partitioned Tables

Siebel Business Applications support two scenarios for converting nonpartitioned tables to partitioned tables.

NOTE: In the following scenarios, 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 Database Server Configuration utility (Migrate Repository and Synchronize 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.

  Database operations do not support rebuilding a table space from nonpartitioned to partitioned. Therefore, if the source table and target table are defined *in the same table space* (for example, if you used the dbconf.xls spreadsheet to convert an existing table space from nonpartitioned to partitioned), you must perform one of the following procedures (as appropriate for your configuration) to convert the nonpartitioned table to a partitioned table.

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 the original table space.
2. Use the Siebel Database Storage Configurator (dbconf.xls) to modify the storage control file, changing the mode of the table spaces to partitioned.
3. Run the Database Server Configuration utility and select the Migrate Repository option. 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

- You 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 applying all changes, 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.

Custom Extensions to the Siebel Schema

There are several different possible 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 "How Siebel Systems Implements LONG Columns on DB2 for z/OS" on page 136.

The process of creating a small extension column is the same on z/OS as on other platforms and is documented in the *Using Siebel Tools* guide.

To create a large extension column or an extension table, you must specify a 16-KB or 32-KB table space:

- If you are developing on DB2 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 143.)
- If you are developing on DB2 for z/OS, you must edit the storage control file to specify the database and table space in which 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.”

Process of Extending the Siebel Schema

To extend the Siebel Schema, perform the following tasks:

1. Check out and lock the project to which the table being extended belongs. In most cases, this is Table Common.

   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 the *Using Siebel Tools* guide.

2. Update the logical schema definition in the development environment using Siebel Tools. This process is the same on z/OS as on other platforms and is documented in the *Using Siebel Tools* and *Developing and Deploying Siebel Business Applications* guides.

   NOTE: Make sure that you select the check box on the Database tab of the Development Tools Options dialog under the View menu. This option validates the objects’ compliance with DB2 for z/OS and sizing conventions.

3. Apply the physical schema extensions to the development database:

   - If you are developing on DB2 for Windows or UNIX, follow the procedures in “Applying Schema Extensions to the Development Database” on page 142.
   - If you are developing on DB2 for z/OS, follow the procedures in “Migrating Customizations from Development to the Target Database” on page 145.

   Some companies do not allow direct database extension. Siebel Tools does not allow the Apply button to be disabled but the Apply process fails if developers do not have appropriate database privileges (such as CREATEDBA and DBADM). For information about the Apply process, see “Applying Schema Changes to the Target Database” on page 148.

   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 145 to apply schema extensions to their development databases.

4 Update and test configuration changes that apply to the extensions.

See the *Using Siebel Tools* and the *Developing and Deploying Siebel Business Applications* guides, the *Applications Administration Guide*, and other documentation on the *Siebel Bookshelf* CD-ROM.

Storage Control File Object Definitions 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 to place the table 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=SIDB6001
Tablespace=S0600100
CLOBS=NO
```

NOTE: If you have different storage control files for your development environment and production environment, you must add the new table object definition to both storage control files.

Applying Schema Extensions to the Development Database

After your customizations are completed, you must update your test environment.

NOTE: Siebel 7 does not support customized database triggers. If you create customized triggers on your Siebel base tables, you must disable them before updating the logical database schema. You can 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	Select one of the following options from the drop-down menu: ■ All. Updates 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. ■ Current Query. Updates the database to reflect modifications made to the tables in the current query only. ■ Current Row. Updates the database to reflect modifications made to the table in the current row only.
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.
Storage control file	Specify the complete path for the storage control file. This file is provided by the DBA and is specific to your database.
Database user	Enter the ID of a database user with CREATEDBA or DBADM privileges. The schema qualifier is read from tools.cfg.
Database user password	Enter the password of a database user with CREATEDBA or DBADM privileges.
ODBC data source	Verify that the ODBC connection specified in the ODBC Data Source text box is correct for your environment. 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).

5 To activate extensions to EIM tables, select the appropriate tables, and then click Activate.

Spooling DDL Generated by the Apply Button

The Siebel environment variable, SIEBEL_GENERATE_DDL, allows you to record into a file the DDL that is generated when you click the Apply button in Siebel Tools.

To spool DDL, generated by clicking the Apply button, into a file

1. Set the environment variable before running Siebel Tools on Windows from the command line or using Windows environment parameters:

   - If using the command line, set `SIEBEL_GENERATE_DDL=Y`.
   - If using the Windows environment parameters, do the following:

   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 box, click New.

   d. The New System Variable window appears.

   e. Enter `SIEBEL_GENERATE_DDL` in the Variable name field and Y in the Variable value field, and click OK.

   f. Continue to click OK until you are out of the System Variable window.

2. Open Siebel Tools, and click the Apply button.

The DDL is generated to the file output.sql, which is found in the *SIEBEL_ROOT*\bin directory (Windows) or the *$SIEBEL_ROOT*/bin directory (UNIX), where *SIEBEL_ROOT* is the directory where Siebel Tools is installed.

Process of Applying Schema Extensions to the Target Database

After testing 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 145
- "Applying Schema Changes to the Target Database" on page 148.

Preparing the Target Database

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, edit the storage control file to specify the database and table spaces where the new extension tables should reside.

- Ask all mobile users to synchronize.
- Make sure all connected clients are disconnected from the Siebel Database Server.
- When 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 contain data in the target database, 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

NOTE: Before you migrate your customizations from your development database to your target database, validate your storage control file. See "Validating the Extracted Storage Control File" on page 106.

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 mode 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 mode to generate the DDL required to create the Siebel Schema into files for transfer to the DB2 host.

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 for z/OS, your DBA can clone the development database to the target database. For more information, see "Process of Cloning a DB2 for z/OS Database" on page 152.

NOTE: Siebel 7 does not support customized database triggers. If you have created customized triggers on your Siebel base tables, disable them before migrating the database schema. You can re-create the triggers after the migration is finished.

To migrate the schema under Windows

1. Launch the Database Server Configuration utility and follow the steps in "Performing a Standard Installation Under Windows" until the Siebel Database Operation screen is displayed (Step 6 on page 86).
2. Select the Migrate Repository option, and then click Next.
3. In the Source Repository Selection Screen, choose one of the following options:
   - Read source repository directly from the database.
   - Read source repository from a previously exported file.

     If you select this option, the Repository File Selection screen is displayed. Accept the default repository migration file displayed or use the Browse button to select an alternate repository file.

   Click Next to continue.

4 Indicate whether the target enterprise will be online or offline when migrations starts, and click Next.

5 Specify whether or not the migration involves changes to the Siebel Schema, and click Next.

6 On the Language Selection screen, indicate the language in which the target database runs, and click Next.

7 Enter the name of the target database repository, for example, Siebel Repository, and click Next.

8 In the Target RDBMS Platform screen, select IBM DB2 for z/OS, and then click Next.

9 Enter the target database ODBC data source name and the SubSystem name, and click Next.

10 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.

11 In the Target Schema Qualifier screen, type the target schema qualifier in uppercase and the target group authorization ID, and then click Next.

12 In the Migrate Repository Mechanism screen, select the Generate DDL Into Files option, and then click Next.

13 On the DDL Commit Frequency screen, choose the number of DDL statements that can be run before a COMMIT statement is issued, and click Next.

14 In the Output Directory screen, enter the DDL output directory and storage control file, or use the Browse button to locate these files.

For example, specify the following for the output directory:

DBSRVR_ROOT\db2390\dboutput\dev2prod

and the following for the storage control file:

DBSRVR_ROOT\db2390\my_storage_file.ctl

Click Next.

15 In the Log Output Directory screen, specify where log files generated during the migration are to be created, for example, *DBSRVR_ROOT*\db2390\dboutput\dev2prod, and click Next to continue.

16 The runnow screen appears, giving you the choice of applying the configuration now or applying it later. To apply the configuration at any time in the future, navigate to the *SIEBSRVR_ROOT*\bin directory and enter the following on the command line:

```
siebupg.exe /m master_dev2prod_mf.ucf
```

17 Click OK and the Configuration Parameter Review screen appears.

18 Review the parameters, and then click Finish.

The Siebel Upgrade Wizard launches.

19 Click OK to perform the migration.

The Siebel Upgrade Wizard creates the following files:

- **ftpsync.txt.** FTPs the files that you are migrating from development to production on the DB2 host.
- **jobsync.txt.** Sets up the development to production migration process on the DB2 host. Use ftpsync.txt to send this job to the mainframe.
- **unload.ldc.** Unloads the control cards that are used to unload tables during the migration from development to production on the DB2 host.
- **load.ldc.** Loads the control cards that are used to load tables during the migration from development to production on the DB2 host.
- **schema.sql.** Applies schema changes from the schema ddl data set during the migration from development to production on the DB2 host.
- **bumpver.sql.** Bumps up the Siebel Schema version during the migration from development to production on the DB2 host.

To migrate the schema under UNIX

1. Launch the Database Server Configuration utility and follow the steps in "Performing a Standard Installation Under UNIX" until the Siebel Database Operation screen is displayed (Step 10 on page 92).
2. Select option 4 - Migrate Repository.
3. On the Source Repository Selection screen, choose one of the following options:
   - Read source repository directly from the database.
   - Read source repository from a previously exported file.

     If you select this option, the Repository File Name screen is displayed. Accept the default repository migration file displayed or enter the path of an alternate repository file.
4. Indicate whether the target enterprise will be online or offline when migrations starts.
5. Specify whether or not the migration involves changes to the Siebel Schema.
6. On the Language Selection screen, indicate the language in which the database runs.
7. Enter the database ODBC data source name.
8. Enter the database username and password.
9. Enter the Siebel schema qualifier in uppercase.
10. Specify the name of the source database repository, or accept the default by pressing ENTER.
11. Specify the target database repository name, or accept the default by pressing ENTER.
12. Specify the target database platform, in this case, select the IBM DB2 for z/OS option.
13. Specify the target database ODBC data source name.
14. Specify the target database DB2 subsystem name.
15. Specify the target database username and password.
16. Enter the target schema qualifier and the target group authorization ID.

17 In the Installation Mechanism screen, select the Generate DDL Into a File option.

18 On the DDL Commit Frequency screen, choose the number of DDL statements that can be run before a COMMIT statement is issued.

19 On the DDL Output Directory screen, enter the DDL output directory name and then the name of the storage control file.

For example, `$`*`DBSRVR_ROOT/`*`db2390/dboutput/dev2prod` and `$`*`DBSRVR_ROOT`*`/db2390/storage.ctl,` respectively.

20 In the Common Parameters screen, specify where log files generated during the migration are to be created, for example, `$`*`DBSRVR_ROOT`*`/db2390/dboutput/dev2prod_mf.`

21 Review the parameter values you entered, then enter 1 (Yes) to run the Siebel Upgrade Wizard (srvrupgwiz).

You receive a message stating that the files with the Siebel Schema modifications have been generated in the DDL Output Directory. You can now apply the files on the DB2 host.

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 DB2 host. Send this job to the mainframe using ftpsync.txt.
- **unload.ldc.** Unloads control cards used to unload tables during the migration from development to production on the DB2 host.
- **load.ldc.** Loads control cards used to load tables during the migration from development to production on the DB2 host.
- **schema.sql.** Applies schema changes from the schema ddl data set during the migration from development to production on the DB2 host.
- **bumpver.sql.** Bumps up the Siebel Schema version during the migration from development to production on the DB2 host.

Applying Schema Changes to the Target Database

After generating the DDL into files, you must use Siebel-provided scripts, your own FTP, or a similar file transfer program to transport the DDL files to the zSeries machine, where you execute it using customary methods, for example, SPUFI.

To apply schema changes to the target database using Siebel scripts

1 Navigate to the output directory (*`DBSRVR_ROOT`*`\db2390` under Windows or *`$DBSRVR_ROOT`*`/db2390` under UNIX) and open the ftpsync.txt file.

2 Edit the ftpsync.txt file using the following information:

a Change *`&IP`* to the IP address of your DB2 host, for example `ZM01`.

b. Change *&Username* to your own user name, for example SADMIN.

c. Change all occurrences of SIEBELQ1 to your own high-level qualifier (HLQ), for example, SADMIN. Save the file.

   The 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 blksize=0

quote site cylinders primary=1 secondary=1
send &directoryPath1/jobsync.txt 'HLQ.SIEBEL.V00'

quote site cylinders primary=10 secondary=2
send &directoryPath1/unload.ldc 'HLQ.SIEBEL.V01'

quote site cylinders primary=10 secondary=2
send &directoryPath1/load.ldc 'HLQ.SIEBEL.V02'

quote site cylinders primary=5 secondary=2
send &directoryPath1/schema.sql 'HLQ.SIEBEL.V03'

quote site cylinders primary=1 secondary=1
send &directoryPath1/bumpver.sql 'HLQ.SIEBEL.V04'
```

3. Execute ftpsync.txt and send the file to the host.

   From the command line enter:

   ```
   ftp -vn < ftpsync.txt > ftpsync.log
   ```

4. Enter the password associated with the user name you specified in the ftpsync.txt file.

   Press any key to continue.

5. The log file ftpsync.log is created in the *DBSRVR_ROOT*\db2390 (Windows) or $DBSRVR_ROOT/db2390 (UNIX) directory.

   The log contains information on the file transfer.

6. When the transfer is successfully completed, log on to the mainframe, and edit the *HLQ*.SIEBEL.V00D2P data set by changing the job card and *HLQ.*

7. Submit the job and, when it is completed, make sure that the return code is 0.

   A new partitioned data set (PDS) is created, *HLQ*.SIEBEL.D2P.EXEC.

8. Execute the following REXX exec to apply the DDL generated for the schema migration process.

   ```
   Execute clist HLQ.SIEBEL.D2P.EXEC(SBLD2P)
   ```

9. Verify the schema changes.

Synchronizing Schema Definitions with the Existing Siebel Schema

To synchronize the Siebel repository database definitions with the existing Siebel Schema database, use the Synchronize Schema Definitions utility. Detailed information about this utility is provided in the *Using Siebel Tools* guide.

On DB2 for z/OS, this process involves the existing database physical layout and the contents of the storage control file. The resulting database storage layout reflects 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 under Windows

1. Launch the Database Server Configuration utility and follow the steps in "Performing a Standard Installation Under Windows" until the Siebel Database Operation screen (Step 6 on page 86) is displayed.
2. Select the Run Database Utilities option.
3. Select the Synchronize Schema Definition option.
4. Enter the ODBC data source name and the source database DB2 SubSystem name.
5. Enter the database user name and password, and then enter the password again to confirm it.
6. Specify the Siebel Schema qualifier in uppercase and the security group ID.
7. Specify the Repository Synchronization Mechanism, in this case, select the Generate DDL into Files option.
8. On the Commit Frequency screen, choose the number of DDL statements that can be run before a COMMIT statement is issued.
9. Enter the name of the output directory where the DDL is to be created and the path and name of the storage control file to use in the synchronization process.
10. Enter the name of the Repository with which the existing Siebel database is to be synchronized.
11. Enter the directory where log files are to be created or accept the default directory, `ddlsync_mf`.
12. When the runnow message appears, indicate whether you want to apply the configuration now or later. Click OK to continue.
13. Review the configuration parameter values. If these values are correct, click Finish to apply the configuration.
14. If you have to cancel the configuration or encounter problems and want to rerun the configuration at a later time, navigate to the *SIEBSRVR_ROOT*`\bin` directory and enter the following from the command line:

```
siebupg.exe /m master_ddlsynch_mf.ucf
```

When you synchronize schema definitions using the Generate DDL into files option, two output files are generated in the default directory, *SIEBEL_ROOT*`\dbsrvr\db2390\dboutput\ddlsync`, after the synchronize schema definition configuration is completed:

- schema.sql
- ddlview.sql

15 Ask your DBA to apply these files.

To synchronize schema definitions under UNIX

1 Launch the Database Server Configuration utility and follow the steps in "Performing a Standard Installation Under UNIX", until the Siebel Database Operation screen is displayed (Step 9 on page 92).

2 Select option 5- Run Database Utilities.

3 Select option 1- Synchronize Schema Definition.

4 On the Language Selection screen, indicate the language in which the database is run.

5 Enter the ODBC data source name and the source DB2 SubSystem name.

6 Enter the database user name and password.

7 Specify the Siebel Schema qualifier in uppercase and the group authorization ID.

8 Specify the repository synchronization mechanism, in this case, select option 1- Generate DDL into a File.

9 On the DDL Commit Frequency screen, choose the number of DDL statements that can be run before a COMMIT statement is issued.

10 Enter the name of the output directory where the DDL is to be created and the path and name of the storage control file to use in the synchronization process.

11 Enter the name of the repository with which the existing Siebel database is to be synchronized.

12 Specify the directory where log files are to be created; the default is `ddlsynch_mf`.

13 Review the configuration parameter values and enter 1 (Yes) to launch the Siebel Upgrade Wizard (srvrupgwiz).

NOTE: If you want to rerun the Upgrade Wizard at a later time, navigate to the $*SIEBSRVR_ROOT*/bin directory and enter:

```
srvrupgwiz /m master_ddlsynch_mf.ucf
```

When you synchronize schema definitions using the Generate DDL into a File option, two output files are generated in the default directory, $*SIEBEL_ROOT*`/dbsrvr/db2390/dboutput/DDLSYNC`, after the synchronize schema definition configuration is completed.

- schema.sql
- ddlview.sql

14 Ask your DBA to apply these files.

Process of Cloning a DB2 for z/OS Database

You can clone your existing Siebel database to a target database. The DB2 system cloning process involves the following tasks:

1. "Extracting a Storage Control File from the DB2 Catalog"

   This involves extracting a storage control file from the existing Siebel Schema.

2. "Generating a DDL File from a Storage Control File"

   Use the extracted storage control file to generate a DDL file that you can apply to create a clone of the Siebel Schema.

Generating a DDL File from a Storage Control File

When you have extracted a storage control file from the existing Siebel database, use this storage control file to generate a DDL file. You can then apply the DDL file to the database to create a Siebel Schema clone.

To generate the DDL file under Windows

1. Launch the Database Server Configuration utility and follow the steps in "Performing a Standard Installation Under Windows" until the Select Installation Type screen is displayed (Step 12 on page 87).
2. Select the Customized Install option, and then click Next.
3. Specify the storage control file you want to use to create the DDL, and click Next.
4. In the Select Installation Mechanism screen, select the Generate DDL Into Files option, and click Next.
5. On the DDL Commit Frequency screen, choose the number of DDL statements that can be run before a COMMIT statement is issued, and click Next.
6. In the Output Directory screen, specify the location of the directory in which the DDL is to be created, for example, *DBSRVR_ROOT*`\db2390\dboutput\install.` Click Next.
7. In the Log Output screen, specify the directory where log files are to be created or accept the default. Click Next to continue.
8. The runnow screen appears, giving you the choice of applying the configuration now or later.

   To apply the configuration later, navigate to the *SIEBSRVR_ROOT*`\bin` directory and enter the following from the command line:

```
siebupg.exe /m master_install_mf.ucf
```

Click Ok to continue and the Configuration Parameter Review screen appears.

9. Review the parameters, and then click Finish.

   When the Siebel Upgrade Wizard launches, click OK to continue.

10. When the DDL file is created, 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
    ```

11. Select Yes.

    The following schema files are created and can be applied using either an optional process provided by Siebel or any other tools used in your company for applying DDL.

    - 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. Launch the Database Server Configuration utility and follow the steps in "Performing a Standard Installation Under UNIX" until the Select Installation Type screen is displayed (Step 16 on page 93).
2. Select option 2- Customized Install.
3. In the Storage File screen, specify the storage control file you want to use to generate the DDL.
4. Select the option 1- Generate DDL Into a File installation mechanism.
5. In the DDL Commit Frequency screen, choose the number of DDL statements that can be run before a COMMIT statement is issued.
6. Specify the DDL output directory, for example, `$`*`DBSRVR_ROOT`*`/db2390/dboutput/install`.
7. Specify where you want the log files to be created on the Common Parameters screen.
8. Review the configuration parameters and then enter 1 (Yes) to launch the Siebel Upgrade Wizard (srvrupgwiz).
9. When the DDL files are created, a message appears stating that the schema.sql and ddlview.sql files have been generated in the DDL Output Directory.
10. Exit from srvrupgwiz. You can now apply the schema.sql and ddlview.sql files on the DB2 host to create a clone of the Siebel Schema.

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.

Data Migration

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 the *Developing and Deploying Siebel Business Applications* guide.

Applying Schema Changes to Other Local Databases

For information on upgrading local databases, see the *Using Siebel Tools* and *Developing and Deploying Siebel Business Applications* guides.

Application Customization Using Assignment and Workflow Rules

You can customize applications using Siebel Assignment Manager and Siebel Workflow, as well as by using Siebel Tools.

You can use Siebel Assignment Manager to create business rules that automatically assign entities, such as opportunities, service requests, or activities, to the most qualified individuals. For more information, see the *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, for example, an opportunity being assigned. For more information, see the *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 the database triggers to identify the records that might match policy conditions.

For information on activating rules, see the *Developing and Deploying Siebel Business Applications* guide.

11 Maintenance Considerations for z/OS

This chapter describes maintenance tasks that you can perform to improve the performance of Siebel Business Applications on DB2 for z/OS after you have completed the Siebel application installation.

This chapter consists of the following topics:

DB2 Statistics for Siebel Business Applications

It is recommended that you apply statistics to Siebel applications. 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.

The RUNSTATS utility and the DSTATS utility are recommended for updating statistics.

About the RUNSTATS Utility

In DB2 for z/OS v8, you can use RUNSTATS to collect distribution and frequency statistics for all types of columns (indexed and nonindexed) and for user-defined groups of columns. You can use RUNSTATs to collect statistics on the most and least frequently occurring values, and to collect cardinal values for groups of columns.

In DB2 for z/OS v7, you can use RUNSTATS to collect 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.

Execution of the rstat390 utility from midtier executes RUNSTATS on the DB2 host. The rstat390 script is located within the *SIEBSRVR_ROOT*\bin directory (Windows) or the *$SIEBSRVR_ROOT*/bin directory (UNIX). For rstat390 help, use option /h.

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, for 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 error does not indicate that your database has been harmed; however, you do have to rerun the RUNSTATS job for any table for which this type of error is 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 option allows concurrent access while the RUNSTATS utility executes. You can also execute RUNSTATS directly from the DB2 host.

When to Use the DSTATS Utility

You can execute RUNSTATS following the reorganization of any fragmented table spaces and indexes. For more information on this topic, see "About Reorganizing Table Spaces, Partitions, and Indexes."

If you are experiencing slow queries, however, run the DSTATS utility. Slow queries can result when the optimizer chooses an inefficient access path as a result of data skew on nonleading indexed columns or nonindexed columns. In such cases, IBM recommends running the DSTATS utility to collect column distribution statistics on these columns; this can lead to significant improvements in 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 further information on DSTATS, 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 40.

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 157.

About Cursor Close

Cursor close allows you to regulate system resource utilization by Siebel clients in a z/OS 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 Developer Web Clients and Siebel Mobile Web Clients.

The Siebel client connector closes the SQL cursor after the number of rows specified by the cursor close parameters, DSMaxCursorSize and DSPreFetchSize, have been retrieved.

You can choose whether to use cursor close in restricted or unrestricted modes.

Unrestricted Mode

When cursor configuration parameters, MaxCursorSize and PreFetchSize, are not specified or are both 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 is specified as 128, the Siebel application retrieves up to 128 rows from the database for a single operation or query. The Siebel application then closes 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.
```

Suppose, for example, that you navigate 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 you try to scroll to the 129th opportunity, the cursor close alert appears. You then have to choose Edit > Query > Refine Query, type `‘>Siebel Systems’` in the Name field, and then execute the query to retrieve additional records.

Cursor Close Parameter Values on DB2 for z/OS

By default, on other DBMS platforms, Siebel Systems sets both DSMaxCursorSize and DSPreFetchSize to a value of -1, which causes the cursors to remain open to retrieve additional data.

When using DB2 for z/OS v7 and v8, setting the values of the DSMaxCursorSize and 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 recommends that zSeries users set the MaxCursorSize and PreFetchSize parameters to positive values.

The minimum acceptable value that you can specify for the Cursor Close parameter is 32. The default value for both MaxCursorSize and PreFetchSize is 128. This value allows a reasonable number of rows to be retrieved without holding open the cursor. After 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 for z/OS. The guidelines outlined above are not mandated, but they provide a balance between system resource utilization, user responsiveness, and productivity. However, your enterprise might have unique requirements beyond the scope of this chapter.

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.

Modifying the Cursor Close Alert Message

You can modify the message that is displayed to the user when there are more rows than can be returned in a query.

To modify the cursor close alert message

1. In your Siebel application, navigate to Site Map > Administration-Application > System Preferences.

   The System Preferences view appears.

2. In the System Preference Name column, select the SSASqlErrRsltsDiscarded parameter.

3. Edit the value in the System Preference Value column to how you want the cursor close alert message to display.

   The maximum message length is 200 characters.

4. Restart the Siebel application.

Changing Cursor Configuration Parameter Values

You can change cursor configuration parameters for Siebel clients in the following locations:

- Application configuration (.cfg) files for the Siebel Developer 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 Developer Web Client

For the Siebel Developer Web Client, you can set cursor configuration parameter values for different classes of users in application configuration files. Table 15 lists the user classes and the recommended cursor configuration parameter values for each class.

Table 15. User Classes and Recommended Cursor Close Parameter Values

User Class	Description	Recommended Value
Siebel administrator	Accesses the system to perform system updates; may need to access large amounts of data.	-1
Siebel application developer	Performs development and customization of the Siebel application using Siebel Tools.	-1
Special	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. Examples include: ■ No requery formulation is available to access the next set of data. ■ Operation requires the complete result set. ■ Data returned must be internally consistent. When a requery is issued, the returned data set might have changed because of other database activity.	-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 Developer 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: You must set MaxCursorSize and PreFetchSize to identical values.

4 Save your changes.

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 15 on page 162.

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 Database Server, and showing how to formulate requeries 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.

4 Save your changes.

Using Siebel Tools to Bypass the Number of Rows Returned

There may be cases where you do not want to change the general limitation concerning the number of rows returned (default 128), but do want to have an unlimited number of rows available for certain operations.

To do this, leave the parameter values for MaxCursorSize and PreFetchSize in the .cfg file set to the default number (or any number you choose), but use Siebel Tools to set values for the specific Business Components for which you want unrestricted operation.

To ignore the Cursor Close limitation, change the properties for the parameters MaxCursorSize and PreFetchSize for these Business Components to -1.

Database Connection Pooling

Siebel 7 introduced Siebel Database Connection Pooling and Multiplexing, a feature that allows database connections to be shared and reused, so reducing the number of DB2 database active threads (DBATs) that are required. 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.

Siebel Database Connection Pooling can give you flexibility in increasing the number of rows returned from a query or in keeping cursors open using the DSMaxCursorSize=-1 and DSPreFetchSize=-1 values. However, you should consider the latter option only if it is essential to your business needs. Using the DSMaxCursorSize=-1 and DSPreFetchSize=-1 settings requires careful DB2 buffer pool tuning and the use of DB2 storage contraction DSNZPARMs. See "About Cursor Close" on page 159 and "DDSNZPARM Parameter Settings for Siebel Applications" on page 34 for further information.

You must also review the unlimited cursor close values in combination with other factors, such as:

- DB2 DBM1 virtual storage
- Transaction rate of arrival
- Average length of a transaction
- Maximum number of connected threads
- Number of expected long-running complex queries

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 only the minimum number remain.

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 additional information on configuring Database Connection Pooling and Multiplexing parameters, see the *Performance Tuning Guide*.

Dynamic SQL in the Siebel Application

Traditional applications are typically based on static, hard-coded SQL. However, Siebel Business Applications generate dynamic SQL statements at run time, using the Siebel Application Object Manager and the program configuration in use by the customer. As a result, SQL statements can be as individual as the deployments in which they occur and, in some cases, dynamic SQL can become very complex, 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.

Advantages of Spooling SQL in the Siebel Application

Using spooling with the Siebel application can be useful for a number of reasons:

- It allows you to identify the SQL statement being executed when an SQL error code is returned.
- It can help you in assessing the time taken by the various SQL statements when you encounter performance issues.
- When data appears to be missing from a view, you can identify the SQL used to retrieve that data.

You can then run the SQL manually, either from the midtier machine, for example using the DB2 Command Center, or on the DB2 for z/OS host, for example using tools such as SPUFI, to see what data is being returned.

To minimize the amount of SQL tracing produced, and, therefore, the size of the log files, limit the tests to the minimum number of screen changes needed to allow the issue to be reproduced. If possible, spool SQL on a developer client as you have more control over the testing which makes it easier to minimize the amount of SQL traced.

Turning on SQL Spooling for the Object Manager and the Database Connector

You can turn on component event logging for the Developer 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 returned 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 identify the most poorly performing queries.

For Developer clients, you can also set these component events as environment variables. They work in addition to the spooling options.

About Tracing SQL Generated by the Siebel Application

The SQL that the Siebel application generates is 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, you may have to trace the relevant SQL statement through one or all of these layers. Contact Siebel support personnel for guidance on the most appropriate tracing for the situation.

Steps to initiate tracing are described in the following topics:

- "Enabling DB2 CLI Tracing"
- "How to Enable ODBC Tracing"
- "Enabling SQL Tracing for Database Utilities"
- "About SQL Tracing on the DB2 Host"

Enabling DB2 CLI Tracing

You can enable DB2 Call Level Interface (CLI) tracing by adding the following lines to the db2cli.ini file (this procedure applies to Windows and UNIX).

To enable DB2 CLI tracing

1. Add a section to the db2cli.ini file labeled [COMMON].
2. Add the following entries to the [COMMON] section.

```
[COMMON]
Trace=1
TracePathName=C:\TRACE
TraceFlush=1
TraceComm=1
```

For each thread, one trace file is created in the path specified in TracePathName (ensure you have already created the path). On UNIX systems, the DB2 process owner must have the appropriate permissions to write to the directory specified in TracePathName.

The resulting trace files are called pid.tid i.e. 382.0 or -23456.0. Only review the trace files created at the time of problem recreation.

Each DB2 process generates its own trace file. The events in the DB2 CLI trace are specific to DB2 clients.

How to Enable ODBC Tracing

You can enable ODBC tracing by following the instructions in FAQ 1343, available on Siebel SupportWeb. ODBC tracing produces results similar to the CLI trace, but focuses on the ODBC API calls. You cannot 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

You can enable SQL tracing by setting this environment variable before you run the Siebel database utilities.

To enable SQL tracing for database utilities

- Set the environment variables as follows:

  ```
  set DBUTIL_LOG_EVENTS=SQLParseAndExecute=5,SQLDBUtilityLog=3
  ```

About SQL Tracing on the DB2 Host

Determine the tracing to set on the DB2 host in conjunction with the DBA. Refer to the documentation for the DB2 performance monitoring tool you use to determine how to enable the tracing.

About the RLF Error Message

You can set the DB2 Resource Limit Facility (RLF) to cancel long-running queries. For more information on using RLF, refer to the appropriate IBM documentation.

When performing long-running queries, users may receive the following RLF error message:

```
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 exceeds the resource limits set on the database server.

If this message results from the execution of a Query by Example (QBE), users should try to simplify the query, or contact their systems administrator and describe what they were doing in the application when the error occurred.

Using odbcsql to Submit SQL Statements

The Siebel Server installation program installs the odbcsql utility in the *SIEBSRVR_ROOT*\bin directory (Windows) or the *$SIEBSRVR_ROOT/*bin directory (UNIX) of the Siebel Server. Siebel Business 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 the DB2 storage group names for data and indexes which you have to specify during the Siebel Database Server installation process.

NOTE: You can also use other utilities that allow you to query and return result sets to the client.

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 (Windows) or the *$SIEBSRVR_ROOT/*bin directory (UNIX) and enter the following from the command prompt:

  ```
  odbcsql /s database_alias /u username /p password
  ```

  for example:

  ```
  odbcsql /s siebsrvr_siebel /u sadmin /p sadmin
  ```

  where:

  /s = the source ODBC DSN for the connection.

  /u = the user ID.

  /p = the password.

If the odbcsql connection is successful, you can enter queries against the DB2 catalog tables. For example, the following query returns a list of all DB2 storage groups:

```
SELECT NAME FROM SYSIBM.SYSSTOGROUP;
```

To exit odbcsql, enter EXIT. For additional information on the utility, enter the following at a command prompt:

```
odbcsql /help
```

The odbcsql utility is also useful in troubleshooting postinstallation connectivity problems that arise in the ODBC layer of your installation.

Enabling DB2 Dynamic Statement Caching

Siebel Business Applications uses the DB2 global statement caching capability. Global statement caching allows dynamic SQL statements that are reexecuted to bypass the PREPARE phase after the first execution. Dynamic statements are prepared once, stored on a global statement cache, then reused many times. The Siebel application also maximizes the reuse of statement caching.

To enable global statement caching either

- Set the CACHEDYN parameter to YES in the macro DSN6SPRM when generating DSNZPARMs
- Set CACHE DYNAMIC SQL to YES when installing DB2.

About Managing the Dynamic Statement Cache

Because DB2 uses the Environmental Descriptor Manager (EDM) component to manage the dynamic statement cache, it is important that you set EDM-related system parameters to appropriate values. Prior to DB2 for z/OS v8, the cached SQL statements were stored in either the primary EDM Pool, or in a data space. In DB2 for z/OS v8, the cached SQL statements are always stored in a dedicated area above the 2GB bar. For guidance on setting the EDM system parameters, see your IBM documentation.

NOTE: Monitor and adjust the EDM Pool and the related cache storage areas based on your workload.

System parameters associated with dynamic statement caching are described in Table 16.

Table 16. System Parameters Associated with Dynamic Statement Caching

Parameter	Value	Description
CACHEDYN	YES	Enables dynamic statement caching.
CONTSTOR	YES	Contracts working storage for connections periodically (every 50 commits or when storage use reaches 2 MB).
EDMDSPAC	10000	Overrides size of data space for EDM pool during statement caching in DB2 for z/OS v7.
EDMPOOL	32768	EDM pool storage size in KB.
EDMSTMTC	102400	Global statement cache size, in KB, used by EDM in DB2 for z/OS v8.
MINSTOR	YES	Minimizes thread storage usage.

How to Invalidate Cached Statements

After DB2 prepares a SQL statement and places it in the global statement cache, it is reused. If you are performing SQL tuning and you want 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.

You can invalidate a dynamically cached statement using one of the following methods:

- Use the RUNSTATS utility to automatically invalidate cached statements that refer to objects against which RUNSTATS is executed. You can run the RUNSTATS utility on any table or table space referenced in the SQL statement.

  In DB2 for z/OS v8, you can use the REPORTNO UPDATE NONE option to invalidate the cache entries without incurring the overhead of executing all the RUNSTATS logic.

- 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 caches. While some ALTER statements require that an object is stopped, others do 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 the named table from the cache.
- Use a CREATE INDEX statement on tables used in the DB2 access path.
- Stop the DB2 subsystem and restart it.

Tracing the Source of a Query

Administrators can identify the terminals from which queries are generated using the Session ID.

The session ID is composed of three parts: host IP in decimal, process ID, and serial number, for example, 172.20.80.25_324_1.

Eventually, the Object Manager and the DB2 host share the same string which can be used to identify the current connection.

After the session ID is displayed at the Server Manager, you can also check it on the database side.

To verify session ID on the 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 appropriate selection to go to the DB2 Command window.
6. At the DB2 Command window, enter the following:

   ```
   -DISPLAY 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. If there are many active threads shown, you should be able to identify 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 for z/OS

The *Global Deployment Guide* contains information on using Coordinated Universal Time (UTC). Since Release 7.x of Siebel Business Applications, the UTC System Preference default value is TRUE so, if you do not want to implement UTC, you must change this value to FALSE.

When installing and configuring UTC on DB2 for z/OS, you should consider the following:

- The UTC upgrader is not supported on DB2 for z/OS.

  Configure UTC using a fresh database install only.

- The Siebel UTC implementation only works if the system time that the database software uses is in UTC.

  Any system or database parameter that implements a time zone different to UTC must remain at zero. The use of any other value is not supported, as it effectively returns the database configuration to a local time zone.

  As an example, the z/OS system parameter, 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 shown:

  ```
  TIMEZONE W.00.00.00   /* GMT */
  ```

12 Migrating Data Using Siebel EIM

This chapter describes special considerations if you are using Siebel Enterprise Integration Manager (EIM) to import, export, update, merge, or delete data within Siebel Business Applications on DB2 for z/OS.

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:

- "About Setting Up EIM for DB2"
- "How to Improve EIM Performance When Importing" on page 176
- "Considerations for EIM Tables" on page 179

About Setting Up EIM for DB2

There are many ways you can optimize EIM data throughput. EIM is similar in function to other DB2 applications, so 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 the clustering U1 index, 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: Do not update the values in the partitioning key because you might have to perform an unload and reload or a REORG on the data to resolve performance issues.

For more information on DB2 partitioning, see Chapter 6, "Understanding Siebel Partitioning."

EIM and 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 "EIM Table Partitioning and Data Distribution" on page 66 for recommendations about how to avoid uneven data distribution in a partitioned table containing data imported by EIM.

Considerations in Planning Data Mapping

Data mapping—an important part of preparing to run EIM—is particularly important on DB2 because of the way DB2 stores data in tables and indexes. When planning your data mapping, take into account such factors as partitioning, lookups during the load phase, and searches for online transactions.

About Importing Data to the Siebel Application for DB2

The Siebel Business Applications Data Model provides for many possible business needs and configurations. Many of the tables, columns, and indexes provided with Siebel Business Applications might not be relevant to your business. Because DB2 stores information in all columns, including NULL columns, consider carefully which data you should import into your Siebel application. Importing 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."

The MS_IDENT Column for DB2 for z/OS

The MS_IDENT column exists in every EIM table and is specific to DB2 for z/OS. 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. Such failure can occur particularly when migrating data from a Siebel application on DB2 for UNIX or Windows to a Siebel application on DB2 for z/OS.

How to Presort Data During Initial Load

When loading the EIM tables with legacy data, sort this data on the partitioning key that is used on the target base table. When sorting string data, use the same character-collating sequence used in the target database. During this load, also preassign and load batch numbers corresponding to the partitioning scheme of the target base tables when possible. This method 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. Preassigning and loading batch numbers generally improves EIM throughput, as the data is then loaded sequentially and by partition into the target base table.

TIP: To further improve performance, after the EIM tables are loaded with partitioned data, unload this data in clustering order, and then reload it.

Optimal EIM Batch Size

You 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 issue 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 timeouts.

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 large-batch testing 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.

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 percent busy. These can either be multiple EIM processes running simultaneously, or the same EIM process repeated against multiple partitions of the same table.

If you run EIM processes in parallel on a DB2 database with the default setting of `UPDATE STATISTICS = TRUE,` deadlocks can occur 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.

DB2 Database Logging Performance

The number of parallel streams that the DB2 for z/OS environment can support is often limited more by the DB2 logging throughput rate than by 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 data sets. The size of the output buffer can be changed using the DSNZPARM parameter, 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 on 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 if a single subsystem cannot handle the logging tasks.

The RUNSTATS Utility

If EIM performance degrades over time, consider running RUNSTATS against the base tables to update the optimizer statistics.

If you set the parameter UPDATE STATISTICS to TRUE in your .ifb file, the EIM process invokes the DSNUTILS stored procedure. This procedure allows EIM to perform RUNSTATS dynamically on temporary columns to obtain optimal statistics numbers in the DB2 catalog. You must install the IBM DSNUTILS stored procedure and enable the z/OS WLM manager.

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 to TRUE only once to collect proper statistics for each given EIM process. As soon as the statistic are collected, reset UPDATE STATISTICS to FALSE for that process.

For more information on RUNSTATS, see “DB2 Statistics for Siebel Business Applications” on page 157.

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).

How to Improve EIM Performance When Importing

There are a number of techniques you can use to improve EIM performance during import processes, particularly during your initial load of data into Siebel Business Applications.

For example, to improve throughput, if certain values in your database are 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.

For further information on improving EIM import performance, see:

- "Tuning the EIM Import Process"
- "How to Improve EIM Performance During Initial Loads" on page 178.

Follow the recommended import order, as described in the *Siebel Enterprise Integration Manager Administration Guide* for both initial and ongoing EIM loads.

Tuning the 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`

   Setting these flags produces a log file containing SQL statements and information about the length of time 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.

   Carefully review 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 the required indexes for EIM and base tables.
5. Run EIM with parameters to reach each partition and repeat single process tuning.
6. Perform .ifb file tuning for each process, and 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.

How to Improve EIM Performance During Initial Loads

There are certain considerations relating to improving EIM performance that are only applicable during an initial load of data into your Siebel application. These are described below.

Unused, Nonunique Indexes

The initial load of data 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.

You can usually improve performance by determining which base table indexes are unused by EIM during the import process and dropping them prior to the initial load. You must later recreate these indexes.

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 must 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 using 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.

The Free Space Parameters

If you use your Siebel application primarily for queries, updates, and deletions, 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.

Insert Performance on Base Tables

You can improve the insert performance on base tables during an initial load of data into your Siebel application by following these guidelines:

- Set PCTFREE to zero for data and sequential index.
- Set FREEPAGE to zero for data and index.
- Use Page Lock if you are inserting many rows or pages sequentially.

- For a data sharing environment, set the member cluster option to reduce space map and data page P-Lock contention.

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 SIDB0401.H0401000 PART 1 PCTFREE 20;
ALTER TABLESPACE SIDB0401.H0401000 PART 2 PCTFREE 20;

ALTER INDEX SIDB0401.S_ADDR_ORG_M6         PCTFREE 20;
ALTER INDEX SIDB0401.S_ADDR_ORG_P1         PCTFREE 20;
ALTER INDEX SIDB0401.S_ADDR_ORG_P99 PART 1 PCTFREE 20;
ALTER INDEX SIDB0401.S_ADDR_ORG_P99 PART 2 PCTFREE 20;
```

Transaction Logging

Siebel transaction logging is unnecessary during an initial EIM load, and should be disabled by setting `LOG TRANSACTIONS = FALSE` in your .ifb 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 have to run other SQL statements first.

Resolving Performance Degradation During the Load Process

When executing the EIM load process, performance might degrade noticeably after a number of batch loads are executed. This performance degradation often results when outdated statistics are loaded on the tables. To resolve the problem, update statistics on the target base tables.

Resolving Errors in the EIM Process

Perform the following steps to resolve errors that can 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 "The RUNSTATS Utility" on page 176 for information about the 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 Business Applications require the setting RETVLCFK=NO. For information on DSNZPARM settings see "DDSNZPARM Parameter Settings for Siebel Applications" on page 34.

- If EIM terminates during a DELETE EXACT process involving deletion of child records, the following error indicates that the maximum allowable locks have been exceeded:

  ```
  EIM Fails - Max Locks Exceeded on DELETE EXACT
  ```

  This termination 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 .ifb file:

  ```
  COMMIT EACH PASS = FALSE

  COMMIT EACH TABLE = FALSE

  ROLLBACK ON ERROR = TRUE
  ```

A Deployment Planning Worksheet

Each time you install a new Siebel Business application using IBM DB2 for z/OS, you must make copies of this worksheet, complete it, and give it to each member of the deployment team. This worksheet includes the following sections:

- "Team Lead Summary"
- "DB2 Connect Information"
- "Siebel Database Server Installation Information" on page 182

Team Lead Summary

Deployment Team Lead:	
Siebel Administrator:	
Privileged User/Siebel Database User:	
DB2 Systems Programmer (SYSADM):	
DB2 Database Administrator (DBADM):	
Security Administrator:	
z/OS System Programmer:	
Midtier System Administrator:	

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