SIEBEL eBusiness 7

Implementing Siebel *e*Business Applications on DB2 UDB for OS/390 and z/OS

Version 7.5

12-FNVK7Z

April 2003

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

Contents

Introduction

Chapter 1. Security

Chapter 2. Preparing for Implementation on the DB2 Host

Chapter 3. Configuring the Database Layout

Chapter 4. Understanding Siebel Partitioning

Chapter 5. Siebel Database Server File Installation at Midtier

Chapter 6. Siebel Schema Installation on the DB2 Host

Chapter 7. Siebel Repository Import, Export, and Postinstallation Tasks

Chapter 8. Customizing a Development Environment

Chapter 9. Maintenance Considerations for zSeries

Chapter 10. Migrating Data Using Siebel EIM

Appendix A. Deployment Planning Worksheet

Appendix B. Optimizing Performance with Buffer Pools

Introduction

This guide provides information, instructions, and guidelines for implementing Siebel eBusiness Applications using IBM DB2 Universal Database (UDB) for OS/390 and z/OS on zSeries machines.

Although job titles and duties at your company might differ from those listed in the following table, the audience for this guide consists primarily of specialists in these categories:

DB2 Database Administrators (DBAs) Persons who administer the IBM DB2 database system, including data loading, system monitoring, backup and recovery, space allocation and sizing, and user account management.

DB2 System Programmers (SYSADM) Persons responsible for DB2 system configuration.

Deployment Team Lead Person responsible for coordinating the deployment.

Midtier System Administrators Persons responsible for administering Siebel Server machines and performing other tasks on midtier systems in the applicable operating environment.

OS/390 and z/OS System Programmers Persons responsible for OS/390 and z/OS system configuration.

Privileged Users Persons whose DB2 database user ID has CREATEDBA or DBADM privileges.

Security Administrators Persons responsible for systems for implementing data and system security solutions.

Siebel Application Administrators Persons responsible for planning, setting up, and maintaining Siebel applications.

Siebel System Administrators Persons responsible for the whole system, including installing, maintaining, and upgrading Siebel applications.

How This Guide Is Organized

This guide is a sequel and companion to *Siebel Server Installation Guide for Microsoft Windows* or *Siebel Server Installation Guide for UNIX*, as appropriate to the platform on which your Siebel Enterprise Server runs. You cannot install Siebel eBusiness Applications without first following the installation instructions in that guide, because installing the Siebel Enterprise Server is a prerequisite to installing your Siebel Database.

This guide covers implementations of Siebel eBusiness Applications on the DB2 UDB for OS/390 and z/OS platforms, including:

- Security concepts, roles, and permissions required to set up a connection to DB2 and to install Siebel eBusiness Applications on DB2
- Creation and optimization of the DB2 subsystem in preparation for installing and configuring the Siebel Schema
- Descriptions of key concepts and utilities used to configure database layout for installation, configuration, and customization
- Guidelines for partitioning large tables
- Siebel Schema installation and configuration (covered in a sequence of three chapters)
- Issues regarding customization of Siebel eBusiness Applications specific to DB2 UDB for OS/390 and z/OS, including migration from a development to a test environment or to a production environment
- Special platform-specific considerations in using Siebel Enterprise Integration Management (EIM) to migrate large amounts of legacy data

A worksheet is provided for planning your deployment.

NOTE: Your Siebel implementation might not have all the features described in this guide, depending on which software licenses you have purchased.

Unless otherwise indicated, any reference in this guide to *DB2* means IBM DB2 Universal Database for the OS/390 and z/OS platforms. DB2 is also referred to as DB2 UDB for OS/390 and z/OS. Unless otherwise indicated, statements that apply to the OS/390 platform also apply to z/OS.

Any IBM DB2 Universal Database reference that applies specifically to the Microsoft Windows or UNIX platforms is explicitly called out.

Additional Resources

The product documentation set for Siebel eBusiness Applications is provided on the *Siebel Bookshelf*.

Siebel Systems, Inc., reserves the right to modify the documentation for Siebel eBusiness Applications at any time. For updates to Siebel documentation, go to the SupportWeb site (http://ebusiness.siebel.com/supportweb/).

To access SupportWeb, you must provide the user name and password you received from Siebel Support Services (support@siebel.com).

Additional Documentation

The following documents (as appropriate for the operating system you are using) contain procedures that you must perform before you can use this guide:

- *Siebel Server Installation Guide for Microsoft Windows*
- *Siebel Server Installation Guide for UNIX*

Revision History

Implementing Siebel eBusiness Applications on DB2 UDB for OS/390 and z/OS, Version 7.5

Security 1

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

For a discussion of Siebel architecture in general, including a discussion of supported client types, refer to *Siebel Web Client Administration Guide*.

Siebel Application Data Security

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

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

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

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

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

For more information on these topics, refer to *Applications Administration Guide* and *Authentication and Access Control Administration Guide*.

DB2 Operating System Security

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

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

The user name can be associated by the OS/390 Security Administrator with a group ID within the OS/390 security package to simplify database security privilege administration.

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

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

Password Change and Expiration

Because of security constraints, zSeries customers periodically need to expire user passwords.

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

Changing Passwords from Remote Clients

In order to change OS/390 and 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 the `DSNZPARM EXTSEC=YES`.

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

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

OS/390 and 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 *DB2 UDB for OS/390 and z/OS Administration Guide*. Access to database resources is established using one or more of the following authorization IDs:

- **Current SQLID.** This is the ID that defaults to the user ID used when logging into DB2. It can be changed by a `SET CURRENT SQLID` command.
- **Secondary Authorization ID.** This ID is generally known as a group ID. Siebel applications effectively use secondary authorization IDs that can be enabled through the DB2 exit DSN3@ATH. IBM provides a sample exit for setting secondary authorization IDs.
- **Specially bound package owners.** The owner ID is used when `DYNAMICRULES(BIND)` is specified on packages used with DB2 Connect.

Using a Group ID

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

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

Grant statements for additional group IDs. Group IDs must be created separately. Siebel applications include the grantstat.sql script to generate grant statements to grant access to interface tables. For a discussion of the grantstat.sql script, see "Granting Table Privileges" on page 199.

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

To disable a grant, issue a revoke statement.

Using an External Security Adapter

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

Using an external security adapter makes administration easier, because you do not have to create an account for each Siebel application user on the OS/390 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.

NOTE: For LDAP, the generic set of user credentials can be the same for every user, if desired.

The following paragraphs summarize the steps required to configure LDAP for use with Siebel eBusiness Applications. For detailed instructions on how to configure LDAP, refer to *Applications Administration Guide* and *Security Guide for Siebel eBusiness Applications*.

The default user objectclass in LDAP is inetOrgPerson (or one of its descendants). The user ID can be stored in the uid attribute, the password in the userpassword attribute, and the database credentials in any unused directory string-type attribute, such as mail.

Database credentials should take the following form:

```
username=db_user password=db_password
```

where:

`db_user` = a valid user ID with appropriate access; it need not be a Siebel user.

`db_password` = the password for the given user ID (in lowercase characters).

NOTE: The values for user name and password must be lowercase. The credential contains one space between the two parameters and no additional spaces.

You must configure LDAP in several different locations, as described below.

To configure LDAP

1 Edit the appropriate Siebel Application Object Managers (AOM):

- **a** Launch Siebel Call Center or another Siebel application and activate Server Manager. For instructions on how to use Server Manager, refer to *Siebel Server Administration Guide*.
- **b** Locate the applicable configuration file for your application (for example, uagent.cfg for Call Center) and supply the following information in the `[LDAP]` section of the file:

Parameter	Description
`ServerName`	LDAP server host name
`Port`	LDAP port (usually 389)
`BaseDN`	Represents the location in LDAP where users are located.
`UsernameAttributeType`	User ID
`PasswordAttributeType`	User password
`CredentialsAttributeType`	Mail

2 Edit the eapps.cfg file of your Siebel Web Server Extension.

   a Locate the eapps.cfg file in the binary subdirectory of your Siebel Web Server Extension installation directory.

   b Open the eapps.cfg file using any text editor, such as Notepad on Windows or vi on UNIX, and edit it appropriately for your environment, using the following example:

```
[/callcenter_enu]

AnonUserName = user1

AnonPassword = password1
```

NOTE: Make sure the parameters `AnonUserName` and `AnonPassword` for the application you are using are valid for a Siebel user that exists in your LDAP server.

3 If your enterprise uses applications such as Siebel eService, and you want users to be able to self-register or you want to use a delegated administrator, you must revise and activate the following workflows:

- User Registration Company Information (SCW)
- User Registration Forgot Password Process
- User Registration Initial Process
- User Registration SubProcess
- User Registration Process
- User Registration Individual Information (SCW)

4 Set the following system OS Preferences:

Parameter	Value	Description
`SecExternalUserAdministration`	`FALSE`	Allows user-administration with LDAP for Siebel eBusiness Applications.
`SecThickClientExtAuthent`	`TRUE`	Allows users to log into a Siebel application through the Siebel Web Client, using LDAP for authentication.
`Security Adapter CRC`	`0` (zero)	Not used in this case. For more information, see the *Siebel Bookshelf* CD-ROM.

5 Configure the applications your enterprise uses for LDAP by setting the `SecurityAdapter` parameter to `LDAP` in their Application Object Manager, using the instructions in Step 1 on page 24. In this example, `sccobjmgr_enu` is the object manager for Siebel Call Center.

- To set all Siebel applications to use LDAP, using the command-line tool `srvrmgr`, enter the following command:

  ```
  change ent param SecurityAdapter=LDAP
  ```

- To change an individual application, enter the following command:

  ```
  change param SecurityAdatper=LDAP for comp YourAppObjMgr_lang
  ```

  where:

  `YourAppObjMgr` is the name of the Application Object Manager that applies, appended by the three-letter language prefix, such as `sccobjmgr_enu` for Siebel Call Center.

Data Transmission Security

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

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

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

Roles and Permissions

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

- SYSADM
- DBADM
- CREATEDBA

SYSADM

A DB2 subsystem is a prerequisite for installing Siebel eBusiness Applications. While installation does not require the use of an ID with SYSADM privileges, the creation of some underlying DB2 resources might. For detailed information on setting up a DB2 subsystem for Siebel eBusiness Applications, see Chapter 2, "Preparing for Implementation on the DB2 Host."

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

- Allocating and accessing buffer pools
- Allocating and accessing storage groups
- Granting CREATEDBA or DBADM to the Siebel installation user ID

DBADM/CREATEDBA

Installation requires access similar to that of DBADM; installers must be able to create Siebel objects and access the necessary utilities. Therefore, it is recommended that you grant CREATEDBA privileges to the group or user IDs that will be used to perform the installation.

Permission to Access the DB2 Catalog

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

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

- Siebel Database Server users require SELECT authority for `SYSIBM.SYSTABLES`.
- Installation or upgrade users require SELECT authority for the following tables:

  ```
  SYSIBM.SYSAUXRELS

  SYSIBM.SYSCOLUMNS

  SYSIBM.SYSDATABASE

  SYSIBM.SYSINDEXES

  SYSIBM.SYSKEYS

  SYSIBM.SYSSTOGROUP

  SYSIBM.SYSTABLESPACE

  SYSIBM.SYSTABLE

  SYSIBM.SYSTABLEPART

  SYSIBM.SYSTRIGGERS
  ```

To grant access

- Use this command:

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

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

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

Permission to Update Catalog Statistics

NOTE: Updating DB2 catalog statistics is optional.

To give appropriate users the authority to update catalog statistics using the utility loadstat.sql, 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
```

For information on using the loadstat.sql utility, see "Loading Statistics" on page 200.

Required Authorizations

Table 2 lists the authorizations that are required by a DB2 database administrator or system programmer to implement Siebel eBusiness Applications on DB2 UDB for OS/390 and z/OS.

Table 2. Authorizations Needed by DB2 Database Administrators

Task	Authorization Required	Task Command Example
Alter a buffer pool.	SYSADM, SYSCTRL, SYSOPR	`ALTER BUFFERPOOL (BP32K1) HPSIZE(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 SIEBDB01 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 H0001004 IN SIEBDB01 USING STOGROUP SIEBEL PRIQTY 720 SECQTY 720  BUFFERPOOL  BP32K1 SEGSIZE 64 COMPRESS YES LOCKSIZE PAGE;`

Table 2. Authorizations Needed by DB2 Database Administrators

Task	Authorization Required	Task Command Example
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`
Modify 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 SIEBDB12.H0059004;`
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 BP1;`

Table 2. Authorizations Needed by DB2 Database Administrators

Task	Authorization Required	Task Command Example
Grant CREATE or PACKADM for stored procedures.	SYSADM, SYSCTRL	`GRANT CREATE ON COLLECTION SIEBINST TO SIEBDBA;`
Grant CREATEIN for triggers.	SYADM, SYSCTRL	`GRANT CREATEIN ON SCHEMA SIEBTO;`
Grant BINDADD.	SYSADM, SYSCTRL	`GRANT BINDADD TO SIEBDBA;`
Grant SELECT on catalog tables.	SYSADM, SYSCTRL	`GRANT SELECT ON SYSIBM.SYSTABLES TO SIEBTO;`

Table 3 lists authorizations that might be needed by users of Siebel eBusiness Applications. Your enterprise might have unique role names it assigns with the authorities listed below. Therefore, sample role names are offered as examples only.

Table 3. 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;`

Table 3. Authorizations Needed by Siebel Application Users

Task	Role	Authorization Required	Task Command Example
Setting the current SQL ID	Schema qualifier group or individual ID	Owns the schema objects (created by DBADM) that will be used during the installation of the Siebel product	`SET CURRENT SQLID='SIEBTO';`
Performing server functions such as: ■ Runstats ■ Generate triggers	Siebel administrator group	■ Member of the generic user role ■ Actual member of the RACF (or other security package) group, selected to act as administrator	`SIEBDBA`

NOTE: SSEROLE and SADMIN are Siebel defaults for the Siebel user group and the Siebel administrator. However, these names are not required and can be changed to meet your organization's naming conventions.

Preparing for Implementation on the DB2 Host 2

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

System Connectivity Architecture

The Siebel eBusiness Applications communicate with IBM DB2 Universal Database for OS/390 and z/OS (running on zSeries servers) through IBM DB2 Connect middleware.

The two editions of DB2 Connect are:

- **DB2 Connect Enterprise Edition (EE).** This edition is installed on a midtier server such as a Siebel Server machine. Siebel Server communicates with this middleware to support database connectivity for users running Siebel Web Client. Optionally, this edition can support database connectivity for multiple individual users running Siebel Dedicated Web Client.
- **DB2 Connect Personal Edition (PE).** This edition is installed on a user's local workstation. It supports database connectivity for an individual user running Siebel Dedicated Web Client on a workstation.

DB2 Connect EE can reside on common connection gateway machines. (Do not confuse the concept of a *gateway* machine with Siebel Gateway.) DB2 Connect EE is used to broker the connections to DB2 for multiple database clients (Siebel Servers or Dedicated Web Clients). See Table 4 on page 37 for DB2 Connect configurations.

NOTE: If you deploy DB2 Connect EE with Siebel Dedicated Web Client, you must also install a DB2 Connect run-time component on your Siebel Dedicated Web Client machines.

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

Connecting to the Database Using DB2 Connect

You install the Siebel Database Server on DB2 UDB for OS/390 and z/OS using either manual execution methods or the Database Server Configuration Utility on a client Siebel Server. (Installation methods are discussed in Chapter 6, "Siebel Schema Installation on the DB2 Host.") Subsequently, you will execute SQL on DB2.

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

- **Method 1.** Install DB2 Connect and configure ODBC to connect to DB2, then use 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 37 to determine the configuration appropriate to the Client your enterprise supports, then follow the instructions in "Setting Up DB2 Connect" on page 39

- **Method 2.** Install TCP/IP and use FTP (or any other file transfer program) to transfer files to DB2, then use a native IBM utility such as SPUFI for file execution. This method is outside the scope of this guide. If you want to execute SQL using IBM native utilities, refer to IBM documentation.

Configuration Options for DB2 Connect

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

Table 4. DB2 Connect Configurations

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

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

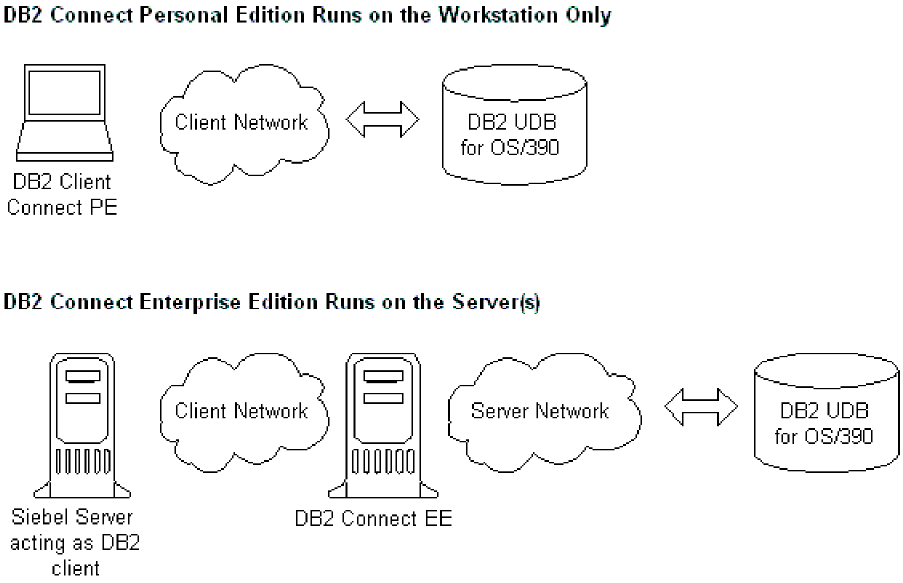

DB2 Connect Personal Edition Runs on the Workstation Only
DB2 Client Connect PE
Client Network
DB2 UDB for OS/390
DB2 Connect Enterprise Edition Runs on the Server(s)
Siebel Server acting as DB2 client
Client Network
DB2 Connect EE
Server Network
DB2 UDB for OS/390

Figure 1. DB2 Connect Enterprise Edition and Personal Edition

For detailed information about DB2 Connect, refer to IBM documentation such as *IBM DB2 Connect User's Guide* or *IBM DB2 Connect Enterprise Edition for UNIX Quick Beginnings*.

Setting Up DB2 Connect

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

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

  To define your database alias, use the DB2 Client Configuration Assistant or the DB2 Command Line Processor. For more information, refer to the IBM document *Administration Guide for DB2 Universal Database for OS/390 and z/OS* (version 7).

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

  To enable ODBC to point to DB2, use the DB2 Client Configuration Assistant’s BIND option to bind the CLI/ODBC Support packages.

- **Install TCP/IP and FTP.** For instructions on installing TCP/IP, refer to the IBM document *Installation Guide for DB2 Universal Database for OS/390 and z/OS* (version 7).

Configuring DB2 Connect EE to Support Siebel Web Client

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

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

This approach is illustrated in the lower part of Figure 1 on page 38.

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

To connect to DB2 using a DB2 Connect EE gateway

1. Using vendor instructions. install DB2 Connect EE on a server machine (typically the machine on which Siebel Server will be installed).

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

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

2. On the DB2 Connect EE Server, upgrade DB2 Connect EE to the appropriate fix pack.

   For information regarding the supported fix pack version, refer to *Siebel System Requirements and Supported Platforms*.

3. On the DB2 Connect EE Server machine, catalog your DB2 OS/390 database, as appropriate, using the DB2 Client Configuration Assistant or the DB2 Command Line Processor.

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

Configuring DB2 Connect EE to Support Siebel Dedicated Web Client

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

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

This configuration enables a high volume of concurrent network transactions between Siebel Dedicated Web Clients and the DB2 Connect EE Server machine. This approach combines elements from both scenarios shown in Figure 1 on page 38.

To connect to DB2 using a DB2 Connect EE gateway

1. Using vendor instructions. install DB2 Connect EE on a server machine (typically the machine on which Siebel Server will be installed).

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

2. On the DB2 Connect EE Server, upgrade DB2 Connect EE to the appropriate fix pack.

   For information regarding the supported fix pack version, refer to *Siebel System Requirements and Supported Platforms*.

3. On the DB2 Connect EE Server machine, catalog your DB2 OS/390 database, as appropriate, using the DB2 Client Configuration Assistant or the DB2 Command Line Processor.

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

Configuring DB2 Connect PE to Support Siebel Dedicated Web Client

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

- Siebel Dedicated Web Client workstations on which you install IBM's DB2 Connect Personal Edition (PE), to act as a gateway to the database.

- DB2 UDB for OS/390 and z/OS

With this approach, users on a Siebel Dedicated Web Client connect directly to DB2. This approach is illustrated in the upper part of Figure 1 on page 38.

To connect to DB2 using DB2 Connect PE

1. Install DB2 Connect PE on each Siebel Dedicated Web Client workstation machine.
2. On each machine, upgrade DB2 Connect PE to the appropriate fix pack.

   For information regarding the supported fix pack version, see *Siebel System Requirements and Supported Platforms*.

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

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

Setting Database Connection Pooling

The IBM DB2 Connect connection concentrator feature (also known as transaction pooling) is not recommended for Siebel 7, because Siebel performance and scalability tests demonstrate no benefits from this feature. The Siebel application provides a similar feature, Siebel DB Connection Pooling, that meets all Siebel performance and scalability requirements on all database platforms. Siebel DB Connection pooling is optimized for Siebel applications.

Siebel Object Manager server components offer the ability to share and reuse database connections, a feature that is referred to as Siebel Database Connection Pooling and Multiplexing. This feature saves the overhead of creating new database connections for each user session and allows the number of concurrent user sessions to exceed the number of database connections.

When you enable Siebel Database Connection Pooling and Multiplexing, the Object Manager component maintains a pool of database connections. When a new connection is needed (a user session is started), the Object Manager looks for an unused connection in the pool. If no connections are available and the pool has less than the maximum number (`MaxSharedDbConns`) of connections, then a new connection is created. When the maximum number of connections is reached, the Object Manager assigns the session to an existing connection and multiplexes the database operations from sessions sharing the connection. When the number of concurrent sessions requires sharing connections, the Object Manager attempts to share the connections equitably. As sessions terminate, the Object Manager keeps track of connection utilization and attempts to maintain the minimum number (`MinSharedDbConns`) of connections in the pool. If there are unused connections, they are released until there are only the minimum number.

Three parameters control Siebel Database Connection Pooling and Multiplexing:

- **MinSharedDbConns.** Minimum number of shared database connections used for general purpose operations.
- **MaxSharedDbConns.** Maximum number of shared database connections used for general purpose operations.
- **MinTrxDbConns.** Minimum number of dedicated database connections used for special purposes.

Set these parameters to the expected maximum number of concurrent users divided by ten (10). This assumes a 30-second think time between operations. You should adjust the ratio in proportion to the actual think time for your particular implementation; for example, if you expect a maximum of 1000 concurrent users, you should set the parameters to `100`.

To disable Database Connection Pooling and Multiplexing, set all three parameters to `-1`. When Database Connection Pooling and Multiplexing is disabled, each user creates a database connection at session creation and releases it at session termination.

Verifying Network Connectivity Under Windows

Before you install the Siebel Database Server, verify your ability to connect to the database across the network.

Verifying the Connection to the Database

Use the Client Configuration Assistant to verify the connection to the database with DB2 Connect.

To verify the connection to the database using the Client Configuration Assistant

1. Launch the Siebel application and choose the database alias, then click test.

   The Connect to DB2 screen appears.

2. Type the user ID and password assigned to you for this database and click OK.

   The following message indicates a successful connection:

   ```
   The connection test was successful.
   ```

Verifying the Connections to the Siebel Gateway, Database Server, and Siebel File System

Use the DB2 Command Window to verify network connectivity to the Siebel Gateway, Database Server, and Siebel File System. In the example below, the command syntax is appropriate for the DB2 Command Window only. It does not work in the DB2 Command Line Processor.

To verify network connectivity using the DB2 Command Window

1 Use a DB2 Command Window to verify connectivity to the database.

a Choose Start > Programs > IBM DB2 > Command Window, and type the following command:

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

where *user_ID* is a valid user name on DB2, and *password* is the appropriate password for that user name.

If the connection is valid, you receive the following message:

```
Database Connection Information
Database Server = DB2/NT x.x.x
SQL authorization ID = SADMIN
Database alias = NAME
```

If the connection is not valid, verify the configuration.

b To close the connection, enter the following command:

```
db2 terminate
```

NOTE: You can also use the DB2 Command Center GUI tool to close the connection.

The Siebel Server installation creates its own ODBC data source.

2 Provide a network connection from the Siebel Server to the Siebel File System, using one of the two following methods:

- Use a universal naming convention (UNC) share.

  For UNC shares, the syntax is `\Siebel_Server\sharename.`

- Connect a shared network drive or device.

  You might elect to map a drive letter (for example, `K:\`) on the computer that is mapped to the Siebel File System directory.

  CAUTION: If this method is used, the drive letter must be permanently mapped to the Siebel File System, and the same drive letter must be mapped on all machines that support a Siebel Server. For this reason, it is recommended that you use a UNC sharename, rather than a shared drive, to connect to the Siebel File System.

3. Check that the Siebel File System directory is visible and that the Siebel Service Owner Account or Resonate Manager Account can copy files to and from it.

   Network connectivity to the Siebel Gateway, Database Server, and Siebel File System is now verified.

The preceding commands should result in the display of the application version and any comments. The display should also indicate whether connectivity through the call-level interface (CLI) layer was set up correctly. Error messages at this point indicate that problems exist downstream from the CLI layer to the database, or that problems exist in the user and password combination or SSEROLE privileges for that user.

Verifying Network Connectivity Under UNIX

This section describes verifying your connectivity on UNIX platforms.

To verify network connectivity

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

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

   where *`user_ID`* is a valid user name on DB2 and *`password`* is the appropriate password for that user name.

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

     ```
     Database Connection Information
     Database server = DB2 OS/390 7.1.1
     SQL authorization ID = SADMIN
     Local database alias = PS01
     ```

   - If the connection is not valid, verify the configuration.
4. To close the connection, enter the following command:

   ```
   db2 terminate
   ```

The preceding commands should result in the display of the application version and any comments. The display should also indicate whether or not connectivity through the call-level interface (CLI) layer was set up correctly. Error messages at this point indicate that problems exist downstream from the CLI layer to the database, or that problems exist in the user/password combination or SSEROLE privileges for that user.

Using Odbcsql to Verify Database Connectivity

The Siebel Server installation program installs a utility called odbcsql in the *SIEBSRVR_ROOT*\bin directory of your Siebel Server. Siebel eBusiness Applications use this program to perform operations on the DB2 database whenever these operations are performed from the midtier—that is, from the Siebel Server.

You can use odbcsql to obtain DB2 storage group names for data and indexes that you will be prompted for during the Siebel Database Server installation process.

It is recommended that you use odbcsql to test your ODBC data source after you install the Siebel Server.

To execute odbcsql

- Enter the following text at a command prompt, as appropriate for your operating system:

  ```
  odbcsql /source database_alias /user username /password password
  ```

  for example:

  ```
  odbcsql /source siebsrvr_siebel /user sadmin /password sadmin
  ```

Using odbcsql, you can then enter queries against the DB2 catalog tables, for example:

```
SELECT NAME FROM SYSIBM.SYSSTOGROUP;
```

This query returns a list of all DB2 storage groups.

To exit odbcsql, enter EXIT.

For additional information, enter the following at a command prompt:

```
odbcsql /help
```

The odbcsql utility is also useful in troubleshooting postinstallation connectivity problems that come from the ODBC layer of your installation.

Installing Required Fix Packs

After you have installed DB2 Connect, you must install any required IBM fix packs. For information on the fix pack required for this release, plus any other installation prerequisites, see *Siebel System Requirements and Supported Platforms*.

NOTE: If a fix pack is installed after the creation of the DB2 database instance, this instance must be updated using the `db2iupdt` command.

Setting Up the Subsystem

Set up a separate subsystem for Siebel eBusiness Applications, particularly for larger deployments, for the following reasons:

- **DSNZPARM optimization.** The DSNZPARMs used with this release of Siebel eBusiness Applications are optimized for the Siebel application, but might not be optimal for use with non-Siebel applications. It is recommended, but not required, that you run the Siebel eBusiness Application on its own subsystem.
- **OLTP and OLAP characteristics.** Siebel applications possess the characteristics of both an OLTP and OLAP application. However, most other vendors' applications have the characteristics of either one or the other.
- **Use of Dynamic SQL.** Virtually all Siebel eBusiness Applications use dynamic SQL, whereas most other applications typically use static SQL and bound plans.
- **DB2 maintenance requirements.** This release of Siebel eBusiness Applications requires the application of a number of maintenance activities that are documented in informational APAR ii13487.

  CAUTION: It is crucial that you apply everything in informational APAR ii13487 to the subsystem that you set up. Otherwise, operating anomalies and unpredictable behavior might occur.

- **System catalog locking during Siebel Database recovery.** 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.

Choosing the Database CCSID

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

Carefully read Release Notes 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

Development databases require ASCII code pages, because databases with EBCDIC code pages do not support two critical procedures in a development environment upgrade:

- Merging prior configuration changes into a new custom configuration repository
- Compiling a new Siebel repository file (.srf file) from the new repository

You can partition table spaces on a database with an EBCDIC code page, but you must modify the storage control file so that limit keys reflect the EBCDIC sort sequence. For more information, see Chapter 8, “Customizing a Development Environment.”

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 DB2 tables in external queries.

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

Setting the CCSID

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

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

Customizing DDF and Setting Up Workload Manager

DDF (Data Distribution Facility) provides a connection between your subsystem and DB2 Connect. You must activate DDF and customize it before you can install Siebel eBusiness Applications.

As part of customizing DDF, you must also configure Workload Manager (WLM). For more information about DDF customization and WLM configuration, see the applicable IBM documentation.

DSNZPARM Parameters for Setting Up the Subsystem

For optimum operation of Siebel eBusiness Applications, use the recommended settings in Table 5 to configure the parameters within DSNZPARM.

The following parameter values are required by Siebel eBusiness Applications:

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

Additional parameter settings in Table 5 are either recommendations or are set by the DBA. You can configure some parameters online, but other parameters require you to shut DB2 down to configure them. For information on which parameters can be updated online, refer to the IBM document *Command Reference for DB2 Universal Database for OS/390 and z/OS (version 7)*.

Table 5. Database Manager Configuration Parameters (DSNZPARM)

Parameter	Explanation	Recommended Setting
DSN6SPRM		
CACHEDYN	Turns on dynamic statement caching. This is a required value for Siebel eBusiness Applications.	YES (required)
CDSSRDEF	Turns off parallelism for dynamic statements. This is a required value for Siebel eBusiness Applications.	1 (required)
CONTSTOR	Compresses storage on a regular basis. Set this value to YES if DBM1 storage is an issue.	YES
EVALUNC	Allows predicate evaluation on uncommitted data. This is a recommended Siebel eBusiness Applications value.	YES
IRLMRWT	Time in seconds before a time-out occurs. This is a recommended Siebel eBusiness Applications value.	300
MINSTOR	Compresses storage on a regular basis. Set this value to YES if DBM1 storage is an issue.	YES

Table 5. Database Manager Configuration Parameters (DSNZPARM)

Parameter	Explanation	Recommended Setting
MXTBJOIN	Sets the maximum number of tables in a join. The default (15) is not sufficient for Siebel eBusiness Applications. NOTE: Unlike other DSNZPARMs that cannot be updated through the installation panels, this parameter must be updated manually.	128 (required)
NPGTHRSH	Allows small tables to use indexes. This is a recommended Siebel eBusiness Applications value.	10
NUMLKUS	Number of locks per user. This value should be monitored and set by the DBA. If you experience persistent locking problems, consider setting the parameter to 0 (unlimited number of locks).	100000
OPTCCOS1	Allows multiple predicate optimizer enhancements.	ON
OPTSUBQ1	Allows non-correlated subquery optimizer enhancements.	YES
PARTKEYU	Allows update of partitioning keys. This is a required Siebel eBusiness Applications value.	YES (required)
RETVLCFK	Allows index-only access of varying-length characters. This value should be set to NO due to padding. This is a recommended Siebel eBusiness Applications value.	NO (required)
DSN6SYSP		
CHKFREQ	Avoids frequent checkpoints in a high-update environment. DBAs should monitor and set this value for between 10 and 20 minutes.	500000 (or 15 when FREQUENCY TYPE = MINUTES)
CONDBAT	Maximum number of concurrent remote connections. The DBA should monitor and set this value.	10000
EXTSEC	Allows DB2 Connect to receive more complete error messages. Allows the changing of passwords from DB2 Connect. This is a required Siebel eBusiness Applications value.	YES (required)
MAXDBAT	Maximum number of database threads (DBAT). The DBA should monitor and set this value.	500

Table 5. Database Manager Configuration Parameters (DSNZPARM)

Parameter	Explanation	Recommended Setting
DSN6FAC		
`CMTSTAT`	Allows a greater number of remote threads without affecting storage. Enables DDF thread pooling. This is a recommended Siebel eBusiness Applications value.	`INACTIVE`
`IDTHTOIN`	Number of seconds before an idle thread is canceled. Prevents long-running threads from holding resources. The DBA should monitor and set this value.	`600`
`POOLINAC`	Number of seconds an inactive thread remains in the DDF pool. The DBA should monitor and set this value.	`120`

Verifying DSNZPARM Settings

DB2 version 7 is shipped with a stored procedure call DSNWZP. This stored procedure returns all the DSNZPARM values for the DB2 on which it is executed. This may be used to compare recommended values with current settings.

NUMLKUS Considerations

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

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

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

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

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

Siebel Dynamic SQL

Traditional applications are typically based on static, hard-coded SQL. However, Siebel eBusiness Applications generate dynamic SQL statements at run time, using the Siebel Application Object Manager and the program configuration in use by the customer. In this way, SQL statements can be as individual as the deployments in which they occur and, in some cases, dynamic SQL might become so complex, that it pushes the limits of DB2. For example, joins with 30-40 tables are not uncommon.

To reduce the complexity of generated SQL (such as the number of joins, the number of columns in SELECT statements, or the number of tables in FROM statements), consider eliminating columns from the standard Siebel applets that you do not require for your business needs, especially in the most frequently used screens and views. You should also keep this consideration in mind, when you customize the Siebel application by creating new objects.

DB2 Dynamic Statement Caching

Siebel eBusiness Applications take advantage of DB2 global statement caching. This allows dynamic SQL statements that are reexecuted to bypass the PREPARE phase after the first execution. Dynamic statement caching allows dynamic statements to be prepared once, stored on a global statement cache, then reused many times. The Siebel application is also designed to maximize the reuse of cached statement caching.

You enabled global statement caching when you set the `CACHEDYN` parameter to `YES` in macro DSN6SPRM when generating DSNZPARMs. Another method of enabling global statement caching is to set Dynamic Cache SQL to `YES` when installing DB2.

Because DB2 saves the cached statement in the Environmental Descriptor Manager (EDM) pool, proper sizing of this pool is very important. For guidance on sizing the EDM pool, see your IBM documentation.

NOTE: The EDM pool should be monitored and adjusted based on your workload.

Other system parameters associated with dynamic statement caching are illustrated in Table 6.

Table 6. System Parameters Associated with Dynamic Statement Caching

Parameter	Value	Description
EDMDSPAC	10,000	Overrides size of data space for EDM pool during statement caching.
EDMPOOL	32	EDM pool storage size.
CONTSTOR	YES	Contracts the EDM pool after 50 commits or > 2 MB in use by one thread.

How to Invalidate Cached Statements

After DB2 prepares a SQL statement and places it in the global statement cache, it gets reused. If a customer is performing SQL tuning and it is desirable to force DB2 to prepare and reoptimize a cached statement, the global SQL statement cache must be invalidated.

To invalidate an entry in the global statement cache for a specific SQL statement, one of the tables referenced by the SQL statement or the table catalog statistics must be altered in some way.

Methods of invalidating a dynamically cached statement include the following:

- Use the RUNSTATS utility to automatically invalidate cached statements that refer to objects against which RUNSTATS was executed. The RUNSTATS utility can be run on any table or table space referenced in the SQL statement.
- Use the DB2 command `STOP` *`OBJECT`* or the SQL DROP, ALTER, or REVOKE statements (for example, `REVOKE` *`ALL`* `FROM PUBLIC`) on any object related to the plan to invalidate both global and local cache. While some ALTER statements might require an object to be stopped, others might not; for example, STOP OBJECT is not required for DROP or REVOKE.
- Use the command `ALTER TABLE` *`tablename`* `AUDIT NONE` to purge statements that reference that table from the cache.
- Use a CREATE INDEX statement on tables used in the DB2 access path.
- Stop the DB2 subsystem and restart it.

Estimating the Storage Space Required

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

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

- Determine the total number, and types, of users of Siebel eBusiness Applications (for example, 500 sales representatives and 75 sales managers).
- Determine the Siebel functionality that you will implement and the entities required to support them. Typically, the largest entities are:
    - Accounts
    - Activities
    - Contacts
    - Forecasts
    - Opportunities
    - Service Requests
- Estimate the average number of entities per user (for example, 100 accounts per sales representative) and calculate an estimated total number of records per entity for your total user base.
- Using standard sizing procedures for your specific database, and the Siebel Schema definition, calculate the average record size per entity and multiply by the total number of records. Typically, these entities span multiple physical tables, all of which must be included in the row size calculation. This calculation determines the estimated data sizes for the largest entities.
- You must add extra space for the storage of other Siebel data. A rough guideline for this amount is one-half the storage required.
- Be sure to allow for a margin of error in your total size calculation.

- Be sure to factor growth rates into your total size calculation.

Allocating Space for Buffer Pools and Storage Groups

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

```
ALTER BUFFERPOOL (BP32K1) VPSIZE (4000)

GRANT USE OF BUFFERPOOL BP32K1 TO PUBLIC;
```

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

```
CREATE STOGROUP SIEBEL VOLUMES ('*') VCAT SIEBEL;

GRANT USE OF STOGROUP SIEBEL TO PUBLIC;
```

Planning the Distribution of Database Objects

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

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

In most implementations, the Siebel tables listed in Table 7 and their corresponding indexes are either the most commonly used tables or the largest tables. For example, the tables S_EVT_ACT, S_PARTY, and S_ORG_EXT are large in all enterprise-level deployments of Siebel eBusiness Applications. These tables and their indexes should be separated across devices. As a general rule, indexes should be on different physical devices from the tables for which they are created.

Table 7. Most Frequently Used and Largest Siebel Tables

Table Names		
S_ACCNT_CHRCTR	S_CON_CHRCTR	S_CYC_CNT_ASSET
S_ACCNT_CO_MSTR	S_CON_TERR	S_DNB_CON_MRC
S_ACCNT_POSTN	S_CRSE_TSTRUN	S_DNB_ORG
S_ADDR_ORG	S_CRSE_TSTRUN_A	S_DNB_ORG_SIC
S_ADDR_PER	S_CS_RUN	S_DNB_UPDATE
S_ASSET	S_CS_RUN_ANSWR	S_DOCK_INIT_ITEM
S_CALL_LST_CON	S_CTLGCAT_PATH	S_DOCK_TXN_LOG
S_DOCK_TXN_LOGT	S_IC_TXN_POSTN	S_INVC_ITM_DTL
S_DOCK_TXN_SET	S_INVOICE	S_INVLOC_ROLLUP
S_DOCK_TXN_SETT	S_INVOICE_ITEM	S_POST_CON
S_ESCL_ACTN_REQ	S_INV_LGR_ENTRY	S_PROC_REQ
S_ESCL_LOG	S_OPTY_POSTN	S_PROD_BASELINE
S_ESCL_REQ	S_OPTY_PROD	S_PROD_CONSUME

Table 7. Most Frequently Used and Largest Siebel Tables

Table Names		
S_EVT_ACT	S_OPTY_TERR	S_PROD_SHIPMENT
S_EXP_ITEM	S_OPTY_POSTN	S_PROD_TARGET
S_EXP_RPT	S_ORG_EXT	S_QUOTE_ITEM
S_EXP_RPT_APPR	S_ORG_TERR	S_SRM_REPLY
S_IC_CALC	S_PARTY	S_SRM_REQUEST
S_IC_CALC_IT	S_PARTY_PER	S_SRM_REQ_PARAM
S_IC_CMPNT_EARN	S_PARTY_REL	S_SRV_REQ
S_IC_TXN	S_PARTY_RPT_REL	
S_IC_TXN_IT	S_POSTN_CON	

Estimating the Number of Database Objects You Need

The number of objects created in the Siebel Schema depends on which Siebel product line you purchase. This consequently determines how much space you must allocate, as illustrated in Table 8. 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 cannot drop unused tables; doing so would cause Siebel Systems to discontinue technical support of your deployment. If you need to drop unused, non-unique Siebel indexes on z/Series, contact Siebel Technical Support for assistance.

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

Table 8. Number of Siebel Database Objects by Product Line

Database Objects	Approximate Number of Objects	
	Horizontal	**Siebel Industry Applications**
Databases containing Siebel tables	18	31
Segmented table spaces containing Siebel tables	150, if grouped, and up to 2,095 if 1 table per table space	254, if grouped, or 2,997 if 1 table per table space
Tables	2,095	2,997
Indexes	10,836	20,727

Configuring the Database Layout 3

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

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

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

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

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

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

About the Siebel Storage Control File Templates

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

The Siebel 7 installation process installs six storage control file templates in the *DBSRVR_ROOT*\DB2390 directory. Select one of the templates that follow based on your partitioning needs, your encoding scheme (if you are using partitioning), and your grouping preference.

The following storage control file templates contain partitioning schemes for a set of tables that use the ASCII encoding scheme:

- storage_p_1to1.ctl — Every nonpartitioned table resides in its own segmented table space.

  Used when you run a Standard database installation with the configuration of one table per table space with partitioning.
- storage_p_group.ctl — Nonpartitioned tables are logically grouped by table space.

  Used when you run a Standard database installation with the configuration of multiple tables per table space with partitioning.

The following storage control file templates contain partitioning schemes for a set of tables that use the EBCDIC encoding scheme:

- storage_p_1to1_e.ctl — Every nonpartitioned table resides in its own segmented table space.

  Used when you run a Standard database installation with the configuration of one table per table space with partitioning.

- storage_p_group_e.ctl Nonpartitioned tables are logically grouped by table space.

  Used when you run a Standard database installation with the configuration of multiple tables per table space with partitioning.

The following storage control file templates contain no partitioning scheme:

- storage_np_1to1.ctl Provides tables in a setup where there is one nonpartitioned table in each table space.

  Used when you run a Standard database installation with the configuration of one table per table space.
- storage_np_group.ctl Provides tables logically grouped by one table space.

  Used when you run a Standard database installation with the configuration of multiple tables per table space.

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

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

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

About Siebel Objects

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

- "Symbolic Variables" on page 69
- "Default Objects" on page 70
- "Buffer Pools" on page 71
- "Storage Groups" on page 72
- "Databases" on page 72
- "Tables" on page 73
- "Table Spaces" on page 75

For an explanation of object naming conventions, see "Object Naming Conventions" on page 76.

Symbolic Variables

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

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

Table 9. Symbolic Variables

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

Default Objects

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   = SIEBDB00
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 you use the apply process without creating new underlying objects for this table in the storage control file, then the necessary storage layout is derived from the default definitions. For information about the apply process, see "Applying Schema Changes to the Target Database" on page 234.

Buffer Pools

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

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

Multiple buffer pools can be used for each page size. If you require multiple buffer pools, use the Siebel Database Storage Configurator (`dbconf.xls`) or any other method described in "Preparing the Database Layout" on page 155 to enter these values.

Your DBA needs to activate and grant use of buffer pools before you can perform the Siebel installation. See "Allocating Space for Buffer Pools and Storage Groups" on page 61 for an example of buffer pool definitions. If you are setting up buffer pools in a production environment, see Appendix B, "Optimizing Performance with Buffer Pools" for a detailed example.

Storage Groups

Siebel eBusiness Applications require one storage group for indexes and one storage group for tables (data). Before performing your installation, ask your DBA to provide you with one storage group name for each. You can 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 61 for an example of storage group definitions.

Databases

Siebel eBusiness Applications ship with up to 24 databases. The number of databases you receive depends on which modules your organization purchased. You can regroup databases or add new databases, as required for your enterprise.

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

```
[Object 4]
Type        = Database
Name        = SIEBDB01
LockSize    = Page
```

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

```
CREATE DATABASE SIEBDB01 CCSID ASCII BUFFERPOOL BP1 /
```

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

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

The names of databases in storage control files are comprised of two parts. The first part of the name is a variable, `$`*`DbnamePrefix`*. The second part of the name is a two-digit number. You can substitute the database name prefix with a literal value up to six characters long (for example, SIEBDB) 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.

The following example illustrates database numbering for certain common databases. It is normal for a gap to exist in database numbering, because the sequence and number of databases you receive depend on the Siebel application that you are implementing.

```
00 – Default
12 – Siebel Seed Tables
15 – DOCKING Tables
16 – Siebel Repository
17 – EIM
18 – Data Warehouse
19 – 22 Verticals
```

Tables

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

```
[Object 2247]
Type       = Table
Name       = S_ORG_EXT
Database   = SIEBDB01
Tablespace = H1465000
Clobs    = No
```

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

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

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

Siebel eBusiness Applications include two templates per encoding scheme for one table per table space and two templates per encoding scheme for multiple tables per table space.

- **One table per table space.** The one-table-per-table space model defines each table in its own table space with no regard to other tables sharing similar functionality. In all other respects these tables are the same as those in the multiple tables per table space template.

- **Multiple tables per table space.** In the multiple tables per table space model, Siebel tables are grouped within databases and table spaces based on functionality. For example, all Activities- or Contacts-related tables are grouped together.

  An advantage of the multiple tables per table space configuration is that fewer underlying physical datasets are created, compared to the one table per table space model. You should consider performance and the granularity of backup and recovery when choosing a template for your deployment.

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

Table 10. Tables with Row-Level Locking

Row-Level Locking
S_ESCL_ACTN_REQ
S_ESCL_LOG
S_ESCL_REQ
S_DOCK_TXN_LOG
S_DOCK_TXN_LOGT
S_DOCK_TXN_SET
S_DOCK_TXN_SETT

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

Table 11 lists tables that use table-level locking.

Table 11. Tables with Table-Level Locking

Table-Level Locking
S_DOCK_INIT_ITEM
S_DOCK_INITM_XX

Table Spaces

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.

Among Siebel templates that contain multiple tables per table space, only one 32-KB table space exists within the Siebel database schema.

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

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

```
[Object 241]
Type       = Tablespace
Name       = H0001004
Database   = SIEBDB01
LockSize   = Page
Bufferpool = BP1
Define     = No
Partitions = 0
```

This Tablespace object translates into the following DDL output:

```
CREATE TABLESPACE H0001004 IN SIEBDB01
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. This is because some values, such as `FREEPAGE` and `PCTFREE`, are taken from the default values.

Object numbering can change when you perform a database 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

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

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

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

Table Space Names

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

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

1. A leading letter (H, V, F, 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).
   - F specifically represents Siebel Financial Services.
   - S represents your custom tables and auxiliary indexes (those that you create for the Siebel application).

2. The four-digit number that was assigned to the base table when you created it, for example, 0401. (This number cascades down to table spaces, auxiliary tables, and auxiliary indexes.)

   NOTE: For custom tables, use a four-digit number greater than 6000.

3. The ending three-digit number depends on the database storage layout and size of the buffer pool. Use the following matrix to determine which prefix to use.

	One table per table space	**Multiple tables per table space**
Partitioned	000	000
Nonpartitioned	000	004 016 032

- If your storage control file is based on the one table per table space model, the suffix is always 000, whether the table space is partitioned or not.
- If your storage control file is based on the multiple tables per table space model:
  - ❑ The suffix for partitioned table spaces is 000.
  - ❑ The suffix for segmented table spaces is 004, 016, or 032, depending on whether you use a 4-KB buffer pool, 16-KB buffer pool, or 32-KB buffer pool.

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

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

Auxiliary Object Names

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

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

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

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

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

NOTE: Default numbering of auxiliary objects starts from 0001.

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

Triggers

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

1. The leading two letters must be PT to designate that this is a partitioning trigger.
2. The third letter (H, V, F, 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).
   - F specifically represents Siebel Financial Services.
   - S represents partitioning triggers used for Assignment Manager or other components.
3. The ending four-digit number uniquely identifies the trigger based on the table number, for example, 0401.

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

About Creating or Modifying the Database Layout

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

- **Method 1. Modify Generated DDL Using DB2 Tools.** Perform a standard installation (see "Performing a Standard Installation" on page 128), and choose the installation mechanism Generate DDL into Files. After the Siebel Upgrade Wizard generates the DDL (to a file named schema.sql), your DBA can directly edit the output DDL using native DB2 tools.

  Some DBAs prefer using Method 1 because it only requires use of customary DB2 tools. This method does not require manipulation of the storage control file.

- **Method 2. 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 (see "Performing a Custom Installation" on page 176) 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.

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

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

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

- Use a Siebel-provided tool called the Siebel Database Storage Configurator (dbconf.xls) to view and edit storage control files and templates. This approach provides flexibility and ease of customization to DBAs who are customizing and managing Siebel DB2 object definitions. For more information about this tool, see "About the Siebel Database Storage Configurator" on page 81.

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

About the Siebel Database Storage Configurator

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

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

The dbconf.xls file includes functions for calculating row count and available space based on row count. The options appear as tabs on the dbconf.xls spreadsheet.

TIP: Append the name of your new storage control file with a suffix; for example, if you used storage_p_1to1.ctl for your input, name the resulting file storage_p_1to1_*suffix*.ctl, where you replace *suffix* with an identifiable character string.

Calculating Space

To calculate space for table spaces and indexes, you need to know your approximate row count. You can use the Row Count spreadsheet in dbconf.xls to enter a list of tables and corresponding row counts.

NOTE: If you already have a valid storage control file (if it already exists and it contains all the values for your configuration), then you do not need to use the import function.

Another option is to create a file that contains a list of tables and row counts. A sample file, row_count.ctl, is delivered with your Siebel eBusiness Application. This file contains the tables and row counts for a fresh install.

The row_count.ctl file only includes tables with a significant number of rows. For all other tables, default sizes are acceptable.

The following example illustrates what row count looks like in the row count spreadsheet.

```
S_ACT_EMP               ,    18319
S_APP_VIEW_RESP         ,   145573
S_APPL_WEB_TMPL         ,    26617
S_APPL_WTMPL_IT         ,   380123
S_APPLET                ,    13836
S_APPLET_INTL           ,    12178
S_AUDIT_ITEM            ,    13405
S_BOCOMP                ,    15380
S_BUSCOMP_UPROP         ,    21078
S_CM_CMD_PARM           ,    13005
S_COLUMN                ,   149196
```

The dbconf.xls spreadsheet provides several functions for calculating space.

- **Import function.** Use the import function to import an existing row count file (you can use the sample file, row_count.ctl). After you update the spreadsheet, use the export function to created a new row count file. Store this spreadsheet in a file that you can use as the input for future space calculations if changes are made.
- **Space calculation.** To calculate space for your installation, use the Import function to import the following files:
  - **Storage control file.** This is the modified version of the storage control file—the version that contains your real values instead of object name variables.
  - **Logical schema.** This is the ddl.ctl file. This information is used to get the index names as well as calculating row length and key length.
  - **Row count file.** This file contains table names and estimated row counts.

The Import function will import the files listed above and then calculate your existing space.

Creating Alter Statements to Adjust Size

The storage control files contain default sizes for indexes and table spaces. You can adjust space during the database storage configuration process. However, another option is to use the Create Alter Statements space calculation built into the dbconf.xls file. The Create Alter Statements function will generate alter table space and alter index statements in a file that can then be executed on the mainframe.

Understanding Siebel Partitioning 4

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

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.

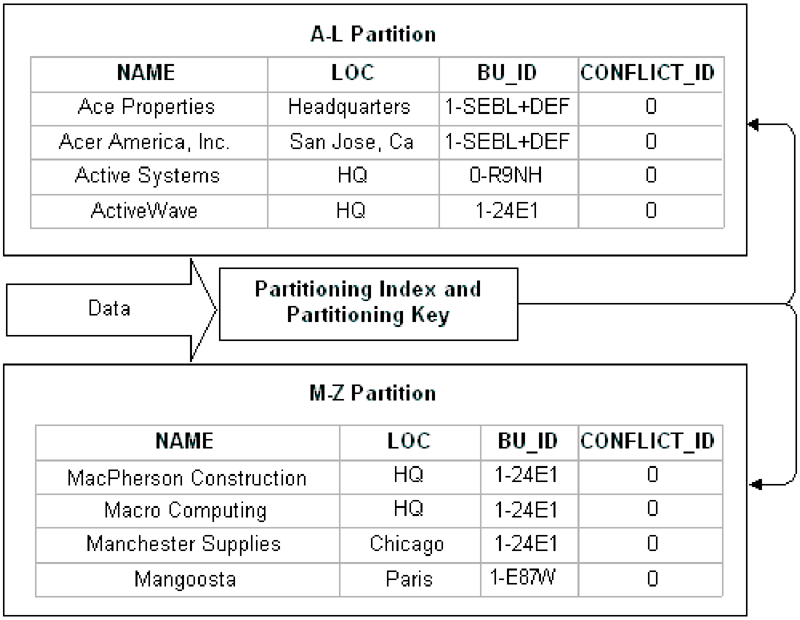
Partitioning EIM table spaces by batch number improves EIM performance for batching and parallel processing, and allows distribution of key ranges across multiple data sets.

Any table, including any of the high-volume tables listed in Table 7 on page 62, 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 103.

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

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

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



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 UDB for OS/390 and z/OS

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

Partitioning Strategy

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

- **storage_p_1to1.ctl.** one table per table space with partitioning for ASCII
- **storage_p_1to1_e.ctl.** one table per table space with partitioning for EBCDIC
- **storage_p_group.ctl.** multiple tables per table space with partitioning for ASCII
- **storage_p_group_e.ctl.** multiple tables per table space with partitioning for EBCDIC

The default partitioning scheme is based on access from the Siebel Client. Storage control files are described in detail in Chapter 3, "Configuring the Database Layout."

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

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

Partitioning Key

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

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

Partitioning Index

The partitioning index is by definition the clustering index in DB2. Therefore, by optimizing your clustering order, you optimize performance. Make sure your partitioning strategy supports more than day-to-day online access to the Siebel tables. Consider processes such as EIM, Siebel Remote, Assignment Manager, and Workflow. See "Strategies for Special Types of Tables" on page 99.

About Configuring Siebel Templates for Partitioning

Four of the storage control templates shipped with your Siebel eBusiness Applications are used for partitioning tables. The templates provide a set of 17 partitioned tables for the Horizontal product line, a set of 24 partitioned tables for Financial Services, and 18 partitioned tables for Siebel Industry Applications.

You can identify the templates for partitioning by the characters _p_ in the middle of the .ctl file name. (If the characters _e_ appear at the end of the file name, then the template is for the EBCDIC code page encoding scheme.) All partitioning templates contain the same Siebel-recommended partitioning schema; the difference between the files is how nonpartitioned tables are stored.

- **storage_p_1to1.ctl.** Contains a partitioning scheme for a set of Siebel tables that use the ASCII encoding scheme. It also contains other nonpartitioned Siebel tables defined as one table per table space.
- **storage_p_1to1_e.ctl.** Contains a partitioning scheme for a set of Siebel tables that use the EBCDIC encoding scheme. It also contains other nonpartitioned Siebel tables defined as one table per table space.
- **storage_p_group.ctl.** Contains a partitioning scheme for a set of Siebel tables that use the ASCII encoding scheme. Most of the nonpartitioned tables are defined as multiple tables per table space.
- **storage_p_group_e.ctl.** Contains a partitioning scheme for a set of Siebel tables that use the EBCDIC encoding scheme. Most of the nonpartitioned tables are defined as multiple tables per table space.

NOTE: Table spaces that contain groups of tables are segmented.

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

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

What to Partition

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

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

For more information about configuring storage control files, see “Preparing the Database Layout” on page 155. For information on how to size your database for small, medium, large, and extra-large installations, refer to IBM’s *DB2 Universal Database for OS/390 Installation Guide*.

Partitioning Methods

Siebel Systems supports two methods for partitioning:

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

Partitioning Based on Business Data

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

Partitioning for Even Data Distribution

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

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

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

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

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

Scenario for Partitioning

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

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

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

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

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

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

The example for object 954 illustrates the Table object definition for S_ADDR_ORG:

```
[Object 954]
Type       = Table
Name       = S_ADDR_ORG
Database   = SIEBDB01
Tablespace = H0401000
Clobs    = No
```

The definition for Tablespace object H0401000 is provided in the example for object 43:

```
[Object 43]
Type       = Tablespace
Name       = H0401000
Database   = SIEBDB01
Define     = No
Partitions = 10
Bufferpool = BP1
LockSize   = Page
```

You can easily identify a partitioned table space by reviewing the Partitions option in the Tablespace object of the storage control file. If the Partitions option is greater than zero, then the table space is partitioned and requires additional objects, such as Tableparts (Tspart) and partitioning index. The number of Tspart objects is equal to the number of partitions. In the example for object 43, illustrated above, Partitions=10; therefore, ten Tspart objects must be defined (as shown in the example for objects 242 through 251).

```
[Object 242]
Type       = Tspart
Name       = H0401000
PartNum    = 1
...
[Object 251]
Type       = Tspart
Name       = H0401000
PartNum    = 10
```

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

```
CREATE TABLESPACE H0401000 IN SIEBDB01 USING STOGROUP SYSDEFLT
PRIQTY 48 SECQTY 1440 FREEPAGE 4 PCTFREE 30 DEFINE NO NUMPARTS 10
(PART 1 USING STOGROUP SYSDEFLT PRIQTY 48 SECQTY 1440 FREEPAGE 4
PCTFREE 30 COMPRESS NO ,
PART 2 USING STOGROUP SYSDEFLT PRIQTY 48 SECQTY 1440 FREEPAGE 4
PCTFREE 30 COMPRESS NO ,
PART 3 USING STOGROUP SYSDEFLT PRIQTY 48 SECQTY 1440 FREEPAGE 4
PCTFREE 30 COMPRESS NO ,
PART 4 USING STOGROUP SYSDEFLT PRIQTY 48 SECQTY 1440 FREEPAGE 4
PCTFREE 30 COMPRESS NO ,
PART 5 USING STOGROUP SYSDEFLT PRIQTY 48 SECQTY 1440 FREEPAGE 4
PCTFREE 30 COMPRESS NO ,
PART 6 USING STOGROUP SYSDEFLT PRIQTY 48 SECQTY 1440 FREEPAGE 4
PCTFREE 30 COMPRESS NO ,
PART 7 USING STOGROUP SYSDEFLT PRIQTY 48 SECQTY 1440 FREEPAGE 4
PCTFREE 30 COMPRESS NO ,
PART 8 USING STOGROUP SYSDEFLT PRIQTY 48 SECQTY 1440 FREEPAGE 4
PCTFREE 30 COMPRESS NO ,
PART 9 USING STOGROUP SYSDEFLT PRIQTY 48 SECQTY 1440 FREEPAGE 4
PCTFREE 30 COMPRESS NO ,
PART 10 USING STOGROUP SYSDEFLT PRIQTY 48 SECQTY 1440 FREEPAGE 4
PCTFREE 30 COMPRESS NO ) BUFFERPOOL BP1 LOCKSIZE PAGE LOCKMAX 0
COMPRESS NO
/
```

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

```
[Object 3198]
Type       = IndexBase
Name       = S_ADDR_ORG_P99
Table      = S_ADDR_ORG
Partitions = 10
SpecialCol = PARTITION_COLUMN WCHAR(2) NOTNULL DEFAULT '  '
Function   = WHEN (N.OU_ID IS NOT NULL) BEGIN ATOMIC SET
N.PARTITION_COLUMN = RIGHT(N.OU_ID, 2); END
Column 1   = PARTITION_COLUMN ASC
Column 2   = OU_ID ASC
```

The example for object 3198 contains a physical partitioning column and a corresponding DB2 before insert trigger. These are always defined by the SpecialCol keyword in an IndexBase section. The syntax for the trigger implements the column PARTITION_COLUMN to S_ADDR_ORG as CHAR(2) NOT NULL with default of a space. It also implements a partitioning index and a trigger to populate the partitioning column.

In the example for object 3198, the IndexBase section defines ten (10) partitions. Therefore, it requires ten IndexPart objects. These are illustrated in the example for objects 3234 through 3243:

```
[Object 3234]
Type       = IndexPart
Name       = S_ADDR_ORG_P99
Table      = S_ADDR_ORG
PartNum    = 1
LimitKey   = 'B'
...
[Object 3243]
Type       = IndexPart
Name       = S_ADDR_ORG_P99
Table      = S_ADDR_ORG
PartNum    = 10
LimitKey   = 'Z'
```

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

```
CREATE  INDEX SIEBTO.S_ADDR_ORG_P99 ON SIEBTO.S_ADDR_ORG
(PARTITION_COLUMN, OU_ID)
USING STOGROUP SYSDEFLT PRIQTY 48 SECQTY 1440 PCTFREE 30 DEFINE NO
CLUSTER (
PART 1 VALUES ('B') USING STOGROUP SYSDEFLT PRIQTY 48 SECQTY 1440
PCTFREE 30 FREEPAGE 4 ,
PART 2 VALUES ('C') USING STOGROUP SYSDEFLT PRIQTY 48 SECQTY 1440
PCTFREE 30 FREEPAGE 4 ,
PART 3 VALUES ('E') USING STOGROUP SYSDEFLT PRIQTY 48 SECQTY 1440
PCTFREE 30 FREEPAGE 4 ,
PART 4 VALUES ('G') USING STOGROUP SYSDEFLT PRIQTY 48 SECQTY 1440
PCTFREE 30 FREEPAGE 4 ,
PART 5 VALUES ('K') 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 ('P') USING STOGROUP SYSDEFLT PRIQTY 48 SECQTY 1440
PCTFREE 30 FREEPAGE 4 ,
PART 8 VALUES ('R') USING STOGROUP SYSDEFLT PRIQTY 48 SECQTY 1440
PCTFREE 30 FREEPAGE 4 ,
PART 9 VALUES ('T') USING STOGROUP SYSDEFLT PRIQTY 48 SECQTY 1440
PCTFREE 30 FREEPAGE 4 ,
PART 10 VALUES ('Z') USING STOGROUP SYSDEFLT PRIQTY 48 SECQTY 1440
PCTFREE 30 FREEPAGE 4
) BUFFERPOOL BP2
/
CREATE TRIGGER SIEBTO.PTH0401 NO CASCADE BEFORE INSERT ON
SIEBTO.S_ADDR_ORG REFERENCING NEW AS N FOR EACH ROW MODE DB2SQL
WHEN (N.OU_ID IS NOT NULL) BEGIN ATOMIC SET N.PARTITION_COLUMN =
RIGHT(N.OU_ID, 2); END
/
```

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

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

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

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

Trigger Performance Overhead

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 parts, 14 columns, and 443 bytes with 8 indexes, have demonstrated a performance overhead of less than one percent per insert.

Changing Partitioning Keys

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

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

```
[Object 4788]
Type        = IndexPart
Name        = S_ADDR_ORG_P99
Table       = S_ADDR_ORG
PartNum     = 1
LimitKey    = 'B'
```

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

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

Strategies for Special Types of Tables

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

Siebel Remote Transaction Logging Table

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

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

To turn off transaction logging using Siebel Tools

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

Performance Considerations for Data Sharing Environment

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

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

Partitioning by MEMBER_NAME is not a default, because 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 7368]
Type       = IndexBase
Name       = S_DOCK_TXN_LOG_PART
Table      = S_DOCK_TXN_LOG
SpecialCol = MEMBER_NAME WCHAR(8)
```

Assignment Manager and Workflow Tables

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

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

- **Hard-coding the actual values for GROUP_ID and RULE_ID.** This approach is recommended when the number of groups is static. Additional partitions can be defined to accommodate new values that are unknown at creation time. To use this option, you must first define the groups and then extract the actual GROUP_ID values you want to use for partitioning.

- **Adding the PARTITION_COLUMN and a trigger to populate the column.** The number of partitions defined should ensure that data from multiple groups goes into multiple partitions. Start with 36 partitions, because this is the base number for the Siebel row ID.

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

EIM Tables

To speed up EIM load and reduce I/O (input and output) contention, make sure all partitions for EIM tables are spread across the entire I/O subsystem. You can use the Siebel Storage Configuration process to define partitions on EIM tables. For detailed information about this process, see Chapter 3, "Configuring the Database Layout."

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

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

Certain EIM tables are already partitioned in the pre-partitioned storage control files (storage_p_1to1.ctl, storage_p_group.ctl, storage_p_1to1_e.ctl, and storage_p_group_e.ctl).

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

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

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

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

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

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


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

EIM Considerations for 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.

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

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

Prepartitioned Siebel Tables

Siebel eBusiness Applications Release 7.0.4 provides the following partitioned tables in storage control file templates.

The tables identified in Table 12 represent partitioned Horizontal product tables; two exceptions are identified as Siebel Industry Applications (SIA) product tables.

Table 12. Prepartitioned Siebel Tables with Physical Columns

Table	Index	Indexed Columns	Trigger Logic

S_ACT_EMP	S_ACT_EMP_P99	PARTITION_COLUMN, EMP_ID	`" WHEN (N.IMP_ID IS NOT NULL)`
			`BEGIN ATOMIC`
			`SET N.PARTITION_COLUMN = RIGHT(N.EMP_ID,2);`
			`END ¬"`

S_ADDR_ORG	S_ADDR_ORG_P99	PARTITION_COLUMN, OU_ID	`" WHEN (N.OU_ID IS NOT NULL)`
			`BEGIN ATOMIC`
			`SET N.PARTITION_COLUMN = RIGHT(N.OU_ID,2);`
			`END ¬"`

S_ADDR_PER	S_ADDR_PER_P99	PARTITION_COLUMN, PER_ID	`" WHEN (N.PER_ID IS NOT NULL)`
			`BEGIN ATOMIC`
			`SET N.PARTITION_COLUMN = RIGHT(N.PER_ID,2);`
			`END ¬"`

S_APPLD_CVRG	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`

Table 12. Prepartitioned Siebel Tables with Physical Columns

Table	Index	Indexed Columns	Trigger Logic

S_ASSET_CON	S_ASSET_CON_P99	PARTITION_COLUMN,ASSET_ID	"WHEN (N.ASSET_ID IS NOT NULL)
			BEGIN ATOMIC
			SET N.PARTITION_COLUMN = RIGHT(N.ASSET_ID,2);
			END ¬"

S_ASSET_POSTN
Siebel Industry Applications (SIA) table
	S_ASSET_POSTN_P99	PARTITION_COLUMN,ASSET_ID	"WHEN (N.ASSET_ID IS NOT NULL)
			BEGIN ATOMIC
			SET N.PARTITION_COLUMN = RIGHT(N.ASSET_ID,2);
			END ¬"

S_COMMUNICATION	S_COMUNICATION_P99	PARTITION_COLUMN,ROW_ID	"WHEN (N.ROW_ID IS NOT NULL)
			BEGIN ATOMIC
			SET N.PARTITION_COLUMN = RIGHT(N.ROW_ID,2);
			END ¬"

S_CONTACT	S_CONTACT_M12	LAST_NAME, FST_NAME, MID_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	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 ¬"

Table 12. Prepartitioned Siebel Tables with Physical Columns

Table	Index	Indexed Columns	Trigger Logic

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 ¬"`

S_INS_CLAIM	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	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	S_INSCLM_ELMNT_P99	PARTITION_COLUMN,INSCLM_ID	`"WHEN (N.INSCLM_ID   IS NOT NULL)`
			`BEGIN ATOMIC`
			`SET N.PARTITION_COLUMN = RIGHT(N.INSCLM_ID,2);`
			`END ¬"`

S_OPTY	S_OPTY_U1	NAME, PR_DEPT_OU_ID, BU_ID, CONFLICT_ID	N/A

S_OPTY_POSTN	S_OPTY_POSTN_P99	PARTITION_COLUMN,OPTY_ID	`"WHEN (N.OPTY_ID     IS NOT NULL)`
			`BEGIN ATOMIC`
			`SET N.PARTITION_COLUMN = RIGHT(N.OPTY_ID,2);`
			`END ¬"`

S_ORG_EXT	S_ORG_EXT_U1	NAME, LOC, BU_ID, CONFLICT_ID	N/A

Table 12. Prepartitioned Siebel Tables with Physical Columns

Table	Index	Indexed Columns	Trigger Logic

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

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 ¬"

Siebel Database Server File Installation at Midtier 5

This chapter describes file installation at midtier on both the Windows and UNIX platforms. For an overview of the involved tasks, see Table 13.

Table 13. Database Server Installation and Configuration Tasks

Who Performs It?	Task
Siebel administrator	**1** Fill out your copy of Appendix A, "Deployment Planning Worksheet" with the parameter values that you need to perform the installation.
	2 Install DB2 Connect. See Chapter 2, "Preparing for Implementation on the DB2 Host."
	3 Install any required fix packs. See *Siebel System Requirements and Supported Platforms*.
Database administrator (DBA)	**4** Create the DB2 subsystem for Siebel eBusiness Applications. See Chapter 2, "Preparing for Implementation on the DB2 Host."
	5 Allocate and configure disk space appropriate to your installation requirements. See also Chapter 2, "Preparing for Implementation on the DB2 Host."
Siebel administrator and DBA	**6** Install the Siebel Database Server software. See "Siebel Database Server Software Installation" on page 111.
	7 Review the software installation. See "Reviewing the Software Installation" on page 121.

About the Siebel Database Server

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

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

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

Preinstallation Tasks

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

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

  For more details regarding authorization requirements, see Chapter 1, "Security."
- A database user (for example, SADMIN) with DBA/CREATEDBA authority exists. This person must also have TSO logon capability for the manual installation process.
- The Database Owner is a valid authorization ID.
- If you have not already done so, make a copy of the Deployment Planning Worksheet, located in Appendix A, "Deployment Planning Worksheet," fill it out, and distribute copies to the entire installation team.

To prepare for the Siebel Schema installation

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

2. Follow the applicable instructions regarding setting up your DB2 Subsystem, as described in Chapter 2, “Preparing for Implementation on the DB2 Host.” This chapter includes installation requirements such as APARs, DB2 work files, disk space, buffer pools, and storage groups.

3. Install DB2 Connect, as described in Chapter 2, “Preparing for Implementation on the DB2 Host,” and test ODBC connectivity between the machine on which the Siebel Server will reside and DB2.

4. Review *Siebel Server Installation Guide for UNIX* or *Siebel Server Installation Guide for Microsoft Windows*, as appropriate to your platform.

5. Install the Siebel Gateway, the Siebel Enterprise Server, and the Siebel Server software in a base language and any additional languages your enterprise requires.

6. Designate a Siebel Server to act as a client to DB2.

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

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

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

Siebel Database Server Software Installation

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

- "Installing the Database Server Software Under Windows"
- "Installing the Database Server Software Under UNIX" on page 117
- "Reviewing the Software Installation" on page 121

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

Installing the Database Server Software Under Windows

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

To install the database server software under Windows

1. Exit any programs that are open.
2. Insert the Windows Server Programs *Language* CD-ROM (where *Language* stands for the Language Pack you want to deploy) into the CD-ROM drive of the chosen machine.

   The machine must have Siebel Server software installed. In the following instructions, the CD-ROM drive is assumed to be D:\.
3. In Windows Explorer, go to the directory `D:\windows_server_`*`LANGUAGE`*`\ses` and double-click `install.exe`.

   If you are installing a U.S. English version of the product, the path appears as `\windows_server_1_enu\ses`.

4. In the Choose Setup Language screen, from the drop-down list, select the language in which you want the installer to appear and click OK.

   If this is the first time you are installing, the Welcome to Servers Setup screen appears. Proceed to Step 6.

   If this is not the first time you are installing, the Existing Installations Found screen appears.

5. In the Existing Installations Found screen:

   - To install a new language pack to an existing installation instance, select it from the screen and click Next. Proceed to Step 11 on page 115.

   - To install this server, leave the displayed installations unselected and click Next. Proceed to Step 6.

6. Read the Welcome to Siebel eBusiness Applications Servers Setup screen and click Next to continue with the setup program.

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

7 In the Select Components screen, choose from the following options:

- Install all components at once by clicking Select All.
- Select all components for which you bought a license.
- Select just the Siebel Database Server at this time for installation and configuration. (You install and configure the other server components individually at a later time.)

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

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

To continue, click Next.

8 In the Setup Type screen, choose the type of Siebel Database Server installation to execute from the following options:

- **Typical.** This setup option installs all Siebel Database Server components except those displayed.
- **Compact.** There is no compact installation option for this server.

- **Custom.** This setup option lets you customize your installation by choosing the specific components you want to install. For example, using the Custom installation option, you can install only the software for DB2 UDB for OS/390 and z/OS, rather than software for all the supported databases.

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

Installation Setup	Products
Typical	Database files, including all the components below: ■ Sample File Attachments ■ MS SQL Server—Database scripts for the Siebel Database on MS SQL Server ■ Oracle—Database scripts for the Siebel Database on Oracle ■ DB2—Database scripts for the Siebel Database on DB2 UDB for Windows and UNIX ■ DB2 for iSeries—Database scripts for the Siebel Database on DB2 UDB for iSeries ■ DB2 for zSeries—Database scripts for the Siebel Schema on DB2 UDB for OS/390 and z/OS
Custom	Select the database files you want to install from the components below: ■ Sample File Attachments ■ MS SQL Server—Database scripts for the Siebel Database on MS SQL Server ■ Oracle—Database scripts for the Siebel Database on Oracle ■ DB2—Database scripts for the Siebel Database on DB2 UDB for Windows and UNIX ■ DB2 for iSeries—Database scripts for the Siebel Database on DB2 UDB for iSeries ■ DB2 for zSeries—Database scripts for the Siebel Schema on DB2 UDB for OS/390 and z/OS

To continue, click Next.

The installer installs Siebel Database Server software in the default directory C:\sea7xx\dbsrvr.

9. If you wish to choose a different installation directory:

   a. Click Browse.

   b. Type the name of the directory you recorded in Appendix A, “Deployment Planning Worksheet.”

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

   ❑ If you chose the Custom installation type, proceed to Step 10.

   ❑ If you chose the Typical installation type, proceed to Step 11.

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

    To continue, click Next.

11. In the Choose Languages screen, select the appropriate Language Pack for the Siebel Database Server. At a minimum, this should consist of the base (primary) server language and (optionally) one or more additional languages.

    You can only select one language at a time.

    NOTE: Unless your database is Unicode-enabled, 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. See *Siebel System Requirements and Supported Platforms* for more information about supported language packs.

    To continue, click Next.

12 When the Select Program Folder screen appears, the installer automatically adds program icons to a folder in the Windows Program file called Siebel Enterprise Server 7.x.x after you click Next.

- To override the default action and install the program icons to a different Program folder in the list under Existing Folders, select that folder and click Next.
- To accept the default action, click Next.

The Start Copying Files screen appears. This screen offers you the opportunity to review the settings you have selected.

13 Review the settings by using the Back button, and when you are satisfied that the values you selected are correct, click Next.

The Setup Status screen appears to show you installation progress.

NOTE: During installation you might be prompted to accept or decline installation of a read-only file. If you receive this prompt, accept it.

14 After all Siebel Database Server files, including the Language Pack you specified, have been installed, the Event Log appears, displaying the status of your installation.

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

To continue, click Next.

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

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

Proceed to “Reviewing the Software Installation” on page 121.

Installing the Database Server Software Under UNIX

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

To install the database server software under UNIX

1 Insert the Solaris|AIX Server Programs *Language* CD-ROM (where *Language* stands for the Language Pack you are installing) into the CD-ROM drive of the chosen computer.

If required by your UNIX configuration, mount the CD.

2 Log on to the server, using the Resonate Manager/Siebel service owner account that you recorded in your copy of Appendix A, "Deployment Planning Worksheet," then enter `ksh` to run a Korn shell. Mount the CD-ROM, if required, to make it accessible.

The volume label for the CD is `sfsaix`*`language2`* or `sfssollanguage2`. The volume label might not be required, depending on how you access the CD-ROM.

3 At the prompt, enter the following command:

```
umask 027
```

This command verifies that the permissions mask for your installation shell is set appropriately.

CAUTION: If you skip this step, the permissions might be set incorrectly on your Siebel application's installation files. This would require you to reset them before continuing with the installation.

4 Start a shell and go to `$`*`SIEBEL_ROOT.`* Depending on the kind of shell you run, enter `. ./siebenv.sh` or `source siebenv.csh` to source environment variables.

NOTE: Make sure there is a space between the two periods in `. ./siebenv.sh`

5 Enter `env` to review environment variable settings, and ensure that the Siebel environment variables *`SIEBEL_ROOT`* (the Siebel file installation directory) and *`LANG`* (the code for the Language Pack you are installing, such as `ENU` for U.S. English) are set correctly to the directory where the Siebel Server software is installed.

NOTE: If you change the location of `$`*`SIEBEL_ROOT,`* you can edit the value here to reflect the new path.

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

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

NOTE: *`LANGUAGE`* is not the same as `LANG`; `LANGUAGE` is the language in which the Configuration Wizard runs, while `LANG` is the code for the Language Pack you installed.

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

csh

```
setenv SIEBEL_ROOT New Value
```

```
setenv LANGUAGE New Value
```

ksh

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

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

6 Go to the following directory on the CD-ROM:

```
dbsrvr/install
```

7. Enter `./install_dbsrvr` to start the Siebel Database Server software installation script.

   The installation script displays the languages available in which to run the installation.

8. Enter the number corresponding to the language in which you want to view installation prompts, for example, `1` for U.S. English (`enu`).

   The script displays the following message:

   ```
   The indicated packages are selected for installation
   [X] IBM DB2 UDB for Unix and Windows
   [ ] IBM DB2 UDB for OS/390 and z/OS
   [ ] Oracle Database Enterprise Edition
   [X] Sample File Attachments
   Would you like to accept these settings? (Y to install files, N
   to change) [Y]
   ```

   The databases displayed are the supported databases for this release.

   - To select a different database, or if you choose not to install the sample file attachments, type `N`.

     NOTE: It is recommended that you select `N`. If you select `Y`, only the DB2 UDB and the sample file attachments will be installed.

     If you select `N`, the installer prompts you separately with the name of each supported database. Proceed to Step 9.

   - To accept the default database and sample file attachments, press ENTER or type `Y`. Proceed to Step 13 on page 120.

9. Indicate your database choices.

   The installer prompts you to select or decline the sample file attachments with your database files.

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

10. Enter N or Y as appropriate.

    The installer displays the values you chose and prompts you to verify your choices.

11. Confirm your choices:

    - To confirm your choices, press ENTER or type Y.

      The installer displays the Language Pack selected for this database server instance.

    - To reject the choices you made, enter N.

      The installer redisplays the database file installation options available.

12. Review the packages selected for installation.

    - To accept the defaults, enter Y at the prompt to proceed with the installation.
    - To make changes to the package selection, enter N at the prompt and make the necessary changes.

13. Confirm your Language Pack choice:

    - To confirm the Language Pack selected, press ENTER or type Y.
    - To abort the installation of this Language Pack, enter N. In this case, you must exit the installation of the database and install the Language Pack you require.

    The database server software installer displays the files it is copying to your disk, creates the required directories and shell scripts, and exits when installation has finished.

    A message informs you when Siebel Database Server file installation is complete.

Reviewing the Software Installation

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

Windows Installation Directory

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

```
DBSRVR_ROOT
          BIN
          COMMON
          DB2390
                         DBOUTPUT\
                                   DDLSYNC
                                   DEV2PROD
                                   INSTALL
                                   UPGRADE
                         STORPROC
                         UPGRADE
          LANGUAGE (for example, ENU)
          FILES
          LOCALE
          PE60
```

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

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

    - **DDLSYNC.** Related to synchronization of the Siebel Schema definition.
    - **DEV2PROD.** Related to migrating development customizations to the test or production environment.
    - **INSTALL.** Contains all manually applied installation files.
    - **UPGRADE.** Related to version upgrade activities.
  - **STORPROC.** Contains stored procedures required for upgrades.
  - **UPGRADE.** Directories containing files to allow upgrade from specific versions of Siebel eBusiness Applications supported for upgrade to the current release.
- ***LANGUAGE.*** Contains language- and database-specific files for the ancestor repository and supporting files. For example, ENU contains U.S. English-language-specific files, and DEU contains German-language-specific files. Additional subdirectories exist under this level, based on the language packs that you install.
- **FILES.** Contains sample file attachments. These should be copied to the appropriate subdirectory within the Siebel File System. See “Populating the Siebel File System” on page 211.
- **LOCALE.** Contains translation files for the Upgrade Wizard. For Siebel Systems use only.
- **PE60.** Files used for uploading data from Siebel Sales Personal Edition.

UNIX Installation Directory

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

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

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

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

    - ❑ **ddlsync.** Related to synchronization of the Siebel Schema definition.

        - **dev2prod.** Related to migrating development customizations to the test or production environment.
        - **install.** Contains all manually applied installation files.
        - **upgrade.** Related to version upgrade activities.
    - **storproc.** Contains stored procedures required for upgrades.
    - **upgrade.** Directories containing files to enable upgrade from specific versions of Siebel eBusiness Applications supported for upgrade to the current release.
- ***language.*** Contains language- and database-specific files for the ancestor repository and supporting files. For example, enu contains U.S. English-language-specific files, and deu contains German-language-specific files. Additional subdirectories exist under this level, based on the language packs that you install.
- **files.** Contains sample file attachments. These should be copied to the appropriate subdirectory within the Siebel File System. See "Populating the Siebel File System" on page 211.
- **locale.** Contains translation files for the Upgrade Wizard. For Siebel Systems use only.

Siebel Schema Installation on the DB2 Host 6

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

For an overview of the tasks for installing the Siebel Schema on the DB2 host, see Table 14.

Table 14. Database Server Installation and Configuration Tasks

Who Performs It?	Task
Siebel administrator and database administrator (DBA)	1 Review information about installing the tables, indexes, and seed data, using either the standard or custom option. See "Installing the Siebel Schema" on page 126. 2 Install the Siebel Schema: ■ If installing by standard mode, see "Performing a Standard Installation" on page 128. ■ If installing by custom mode, see "Preparing the Database Layout" on page 155. Perform the custom installation using the created storage control file. For information, see "Performing a Custom Installation" on page 176. 3 Review the log files for any errors. See "Reviewing the Log Files for Windows" on page 180 or "Reviewing the Log Files for UNIX" on page 184.

Installing the Siebel Schema

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

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

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

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

- **Standard mode.** If you are performing a standard installation, you perform all installation tasks using the Siebel Software Configuration Utility—DB Server Configuration.
- **Custom mode.** If you are performing a custom installation, transfer and execution of the generated DDL are manual processes. Once manual installation of the DDL is complete, the Siebel Software Configuration Utility—DB Server Configuration tool imports the Siebel repository and seed data.

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

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 > Add a language to existing database.

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

Performing a Standard Installation

This procedure describes how to perform a standard installation. You can use the standard installation option as your sole installation procedure, or you can perform the standard installation as a first step of a custom installation and then customize your deployment after you have completed the standard installation. To perform a custom installation, see “Performing a Custom Installation” on page 176.

The standard installation provides six schema layout options. You must select one of the Siebel Systems-provided templates listed below to implement the appropriate layout for your database schema.

Schema Layout	Template Name	Description
One table per table space with partitioning for ASCII	`storage_p_1to1.ctl`	Contains a partitioning scheme for a set of tables; every nonpartitioned table resides in its own segmented table space.
Multiple tables per table space with partitioning for ASCII	`storage_p_group.ctl`	Contains a partitioning scheme for a set of tables; nonpartitioned tables are logically grouped by table space.
One table per table space with partitioning for EBCDIC	`storage_p_1to1_e.ctl`	Contains a partitioning scheme for a set of tables; every nonpartitioned table resides in its own segmented table space.
Multiple tables per table space with partitioning for EBCDIC	`storage_p_group_e.ctl`	Contains a partitioning scheme for a set of tables; nonpartitioned tables are logically grouped by table space.
One table per table space	`storage_np_1to1.ctl`	Contains no partitioning scheme; provides tables in a configuration where there is one nonpartitioned table in each table space.
Multiple tables per table space	`storage_np_group.ctl`	Contains no partitioning scheme; provides tables logically grouped by one table space.

See “About the Siebel Storage Control File Templates” on page 66 for descriptions of the templates.

NOTE: If you choose the Siebel partitioned layout, a select subset of Siebel tables are defined as partitioned with predefined partitioning indexes. For a discussion of partitioned tables and partitioning indexes, see “About Siebel Partitioning” on page 86.

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

Performing a Standard Installation Under Windows

Perform the following steps for a standard installation under Windows.

To perform a standard installation under Windows

1. Launch the Database Server Configuration Utility using any method described in the *Siebel Server Installation Guide for Microsoft Windows*.

   The Gateway Server Address screen appears.

2. Enter the values for the Gateway Server Address and Enterprise Server name as you recorded them in your copy of Appendix A, “Deployment Planning Worksheet.”

   - **Gateway Server Address.** The alias of the host on which you installed the Siebel Gateway.

- **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 Server Address and Enterprise Server Name fields blank.

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

- Accept the default value displayed in the Siebel Server Directory field. (This is the *`SIEBSRVR_ROOT`* directory, for example, `D:\sea7xx\siebsrvr`.)
- Use the browse button to select an alternate directory path.

To continue, click Next.

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

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

To continue, click Next.

5 In the Database Platform screen, select the database platform you use, in this case, DB2 UDB for OS/390 and z/OS.

To continue, click Next.

6 In the Siebel Database Operation screen, select Install Database.

To continue, click Next.

7 In the Select Installation Operation screen:

- If you are configuring Siebel Database Server software for the first time, click Install Siebel Database.

- If you have already installed your Siebel Database Server software in its base or primary language, and want to install a new language, proceed to "Installing Multilingual Seed Data" on page 202.
- Install Siebel Data Warehouse Database. For information on this installation option, see *Siebel Data Warehouse Installation and Administration Guide*.

To continue, click Next.

8 In the ODBC Data Source Name screen, accept the default name of the ODBC data source, `SiebSrvr_Siebel`, or enter the database alias you prefer to use for the data source.

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

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

NOTE: There can only be one database server in a Siebel Enterprise Server.

The Siebel Server installation process automatically creates the data source, using the format `SiebSrvr_`*`enterprise_server_name`*

To continue, click Next.

9 In the Database User Name screen, indicate the following values for your database.

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

- **Database Password (confirm).** Retype the password to confirm it.

To continue, click Next.

10 In the Siebel Schema Qualifier screen, type the up to eight-character ID that identifies the Siebel Schema owner, for example, SIEBTO.

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

11 In the Security Group ID screen, 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 1, "Security."

To continue, click Next.

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

- **Standard Install.** The standard installation option installs the database, using one of the Siebel-recommended default storage layouts.
- **Custom Install.** The custom installation option allows you to install the database using your customized storage layout. See "Performing a Custom Installation" on page 176 for a description of the custom installation process.

To continue, click Next.

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

13 In the Select Installation Mechanism screen, indicate which installation mechanism you want to use.

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

  Use FTP or a similar file transfer program to transport the DDL (schema.sql and ddlview.sql) to DB2, where the DBA executes it, using customary methods, for example, SPUFI or DSNTEP2.

  To continue, click Next and proceed to Step 14 on page 134.

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

  To continue, click Next and proceed to Step 16 on page 134.

14 In the DDL Output Directory screen, select the directory in which you want to store the generated DDL.

- Accept the default directory or use the Browse button, if needed, to locate and map to the directory you have already created for this purpose. The default directory is in the format:

  DBSRVR_ROOT `\db2390\dboutput\install`

- If the directory does not already exist, a message box appears, prompting you that a directory does not exist, click OK. You must now create one.

15 In the Encoding Scheme screen, indicate whether your DB2 subsystem is ASCII or EBCDIC. For more information about choosing the code page for your subsystem, see Chapter 2, “Preparing for Implementation on the DB2 Host.”

16 In the Siebel Schema Layout screen, select one of the following layout options.

- **Multiple Tables per Table Space.** Choose this option if you want to allow multiple tables within a segmented table space.

- **One Table per Table Space.** Choose this option if you want only one table per segmented table space.

- **Multiple Tables per Table Space with Partitioning.** This layout includes a set of tables that is recommended for partitioning. The remaining nonpartitioned tables are arranged in groups within segmented table spaces. For more information on this template, see “About Configuring Siebel Templates for Partitioning” on page 90.

- **One Table per Table Space with Partitioning.** Choose this option if you want a single table per segmented table space. This layout includes a set of tables that is recommended for partitioning. For more information on this template, see “About Configuring Siebel Templates for Partitioning” on page 90.

For details about layout considerations, see Chapter 3, “Configuring the Database Layout.”

To continue, click Next.

17 In the Storage Group for Table Spaces screen, indicate the values for the following parameters.

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

To continue, click Next.

18 In the 4-KB Buffer Pool Name screen, indicate the values for the following parameters.

- **4-KB 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.
- **16-KB 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.

To continue, click Next.

19 In the 32-KB Buffer Pool Name screen, indicate the values for the following parameters.

- **32-KB Buffer Pool Name.** Type the 32-KB buffer pool name for your table spaces, or accept the default, BP32K1. This buffer pool should already be activated and have access to it granted by the DBA.
- **Index Buffer Pool Name.** Type the buffer pool name for the indexes, or accept the default, BP2. This buffer pool should already be activated and have access to it granted by the DBA.

To continue, click Next.

20 In the Database Name Prefix screen, type the prefix for your database name. The default is SIEBDB.

This prefix consists of up to six of the first characters in the names of your 18-22 logical Siebel Databases. This prefix must start with a letter and cannot contain any special characters. All database names end in numbers except for TM, which is used as a temporary database during an upgrade.

To continue, click Next.

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

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

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

  The following message appears:

  ```
  To apply the configuration now, press “OK”.
  To apply the configuration later, press “Cancel”.
  The command line to apply the configuration later is
  C:\sea704\siebsrvr\bin\siebupg \m master_install_mf.ucf
  ```

  NOTE: The last line in the prompt is generated dynamically and can vary based on the location of your Siebel Server installation.

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

```
siebupg \m master_install_mf.ucf
```

Click OK.

23 When the Database Server Configuration Utility displays the items it will execute or import, click OK to start generating DDL.

This step completes the installation option Run DDL Automatically.

If in Step 13 on page 132 you chose the installation option Generate DDL Into Files, the utility displays the following message after it generates the schema.sql file that contains the DDL to create the Siebel Schema:

```
Please create Siebel Schema using schema.sql and ddlview.sql
located in the DDL Output Directory. Once the views are created,
please select Yes. To stop now and continue installation later
please select No. (To resume at a later time, please start the
Upgrade Wizard from the command line using option /m
master_install_mf.ucf).
```

24 Indicate the following:

- To discontinue Siebel Schema (schema.sql) and view (ddlview.sql) generation, click No.

  When you click No, the Database Server Configuration Utility quits. To resume operation of the Upgrade Wizard from the command line, go to `siebsrvr\bin` and enter `siebupg /m master_install_mf.ucf`.

- To continue, click Yes.

  When you click Yes, the utility generates schema.sql and ddlview.sql, which contains the DDL for views. Proceed to Step 25 on page 137.

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

For information on how the DBA applies the DDL on the DB2 host, see “Applying DDL on the DB2 Host” on page 148.

26 If you performed a manual install, validate that all tables and indexes are present in the logical schema. For information, refer to the information in “Validating Siebel Schema Creation” on page 152.

27 Indicate which action you want to take:

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

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

- If you want to quit the DB Server Configuration Utility to give your DBA an opportunity to create the Siebel Schema, click `No`. You can relaunch the wizard at any time to continue this step by submitting the following command from the `siebsrvr\bin` subdirectory:

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

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

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

The following files are generated by the installation into the default output directory (*DBSRVR_ROOT* \db2390\dboutput\install) or into the output directory that you designated in Step 14 on page 134.

- **schema.sql.** Contains DDL for all Siebel DB2 objects. You will transfer this file to the mainframe.
- **ddlsplit.txt.** Contains a REXX routine that can be used to split the DDL into more manageable pieces. This file must be moved to a PDS with RECFM = 80 and a LRECL of 80. You will transfer this file to the mainframe.
- **ddlview.sql.** Contains DDL for all Siebel views. You will transfer this file to the mainframe.

- **instddl.txt.** You will transfer this file to the mainframe. Contains sample JCL to execute the DDL using DSNTEP2. This file must be moved to a PDS with RECFM = 80 and a LRECL of 80. You will transfer this file to the mainframe.
- **instftp.txt.** Use this file to transfer files to the mainframe.
- **schmvld.sql.** Schema validation file
- **seedvld.sql.** Seed validation file containing update statements that gather actual row counts for seed tables.

Performing a Standard Installation Under UNIX

Perform the following steps for a standard installation under UNIX.

To perform a standard installation under UNIX

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

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

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

NOTE: *`LANGUAGE`* is not the same as *`LANG`*`;` the former is the language in which the Configuration Wizard runs, while the latter is the code for the Language Pack you installed.

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

csh

```
setenv SIEBEL_ROOT New Value
```

```
setenv LANGUAGE New Value
```

ksh

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

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

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

```
dbsrvr_config.ksh
```

This launches the Database Server Configuration Wizard.

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

   `SIEBEL_ROOT` = path of your Siebel installation directory

   `LANGUAGE` = language in which you are installing

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

4. Verify and accept the default for the path of your Siebel Server root directory and press ENTER.

5. Verify and accept the default for the path of your database server root directory and press ENTER.

6. Enter the number corresponding to your RDBMS platform, in this case, DB2 for OS/390 and z/OS `(2)`.

7. From the Siebel Database Operations prompt, choose Install Database (`1`).

8. From the Installation Operations prompt, choose Install Database `(1)`.

   - For installation of an additional database language (`2`), see "Installing Multilingual Seed Data" on page 202.

   - For installation instructions for Siebel Data Warehouse Database `(3)`, see *Siebel Data Warehouse Installation and Administration Guide*.

9. Enter the number corresponding with the language in which you want to run your database. (This is known as your base language.) If you are installing your first language, this is (`1`).

10 In the ODBC Data Source Name screen, 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 alias should have been previously registered within DB2 Connect, either by using the DB2 Client Configuration assistant or by natively registering to the DB2 Connect catalog, the Database, Node, and DCS entries.

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

The Siebel Server installation process automatically creates the data source, using the format `SiebSrvr_`*`enterprise_server_name`*.

11 Specify the Database User Name.

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

12 Enter the database password for the Database User Name from Step 11.

13 Specify the Schema Qualifier, for example, SIEBTO.

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

14 Specify the group user ID or accept the default (SSEROLE) by pressing `ENTER`.

This is the group to whom schema access is being granted.

15 Specify the type of installation you want to perform (in this case, `1`):

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

`2 - Custom Install.` The custom installation option allows you to install the database using your customized storage layout. See “Performing a Custom Installation” on page 176 for a description of the custom installation process.

16 Enter the number corresponding to the mechanism you want to use for installation:

`1 - Generate DDL Into Files`. Generates DDL required to create the Siebel Schema. Proceed to Step 17.

You must later use FTP or a similar file transfer program to transport the DDL (`schema.sql` and `ddlview.sql`) to the DB2 host, where the DBA executes it, using customary methods, for example, SPUFI or DSNTEP2.

`2 - Run DDL automatically`. Select this to apply the DDL and DML required to create the Siebel Schema directly against the database. The Siebel application uses ODBC to apply the required DDL to the DB2 host. Proceed to Step 19.

17 Enter the path to the output directory for the DDL files that you will transfer to your zSeries machine, or accept the default directory:

```
dbsrvr/db2390/dboutput/install
```

NOTE: If you plan to use a different output directory to hold the DDL than the Siebel default, you must create it before selecting this option.

18 Enter the number that corresponds to the code page encoding scheme for your Siebel Database:

```
1- ASCII
```

```
2- EBCDIC
```

For more information on choosing the database code page, see "Choosing the Database CCSID" on page 52.

19 Enter the number corresponding to the Siebel Schema Layout you prefer:

```
1 - Multiple Tables per Table Space
```

```
2 - One Table per Table Space
```

```
3 - Multiple Tables per Table Space with Partitioning
```

```
4 - One Table per Table Space with Partitioning
```

For details about layout considerations, see Chapter 3, “Configuring the Database Layout.”

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

The name used should be discussed with the DBA before you start the installation.

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

22 Enter the name for your 4-KB buffer pool, or accept the default BP1.

23 Enter the name for your 16-KB buffer pool, or accept the default, BP16K1.

24 Enter the name for your 32-KB buffer pool, or accept the default, BP32K1.

25 Enter the name for your index buffer pool, or accept the default, BP2.

NOTE: These buffer pools should be activated and granted by the DBA prior to the installation.

26 Enter the Database Name Prefix. The default is `SIEBDB`.

This prefix consists of up to six of the first characters in the names of your 18-22 logical Siebel Databases. This prefix must start with a letter and cannot contain any special characters.

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

28 Review the values for accuracy.

- If the values are correct, launch the Siebel Upgrade Wizard (srvrupgwiz) by entering Y.
- To change your previous values enter N, and rerun the Configuration Wizard by entering `dbsrvr_config.ksh`. This allows you to reconfigure with different values.

This step completes the installation option to Run DDL Automatically.

If in Step 16 on page 143 you chose installation option, `1 - Generate DDL Into Files`, the utility displays the following message after it generates the schema.sql file that contains the DDL to create the Siebel Schema:

```
Please create Siebel Schema using schema.sql and ddlview.sql
located in the DDL Output Directory. Once the views are created,
please select Yes. To stop now and continue installation later
please select No. (To resume at a later time, please start the
Upgrade Wizard from the command line using option /m
master_install_mf.ucf).
```

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

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

29 Indicate the following:

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

  When you click No, the Database Server Configuration Utility quits.

- To continue, click Yes.

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

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

For information on how the DBA applies the DDL on the DB2 host, see "Applying DDL on the DB2 Host" on page 148.

31 If you performed a manual install, validate that all tables and indexes are present in the logical schema. For information, refer to the information in "Validating Siebel Schema Creation" on page 152.

32 Indicate which action you want to take:

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

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

- If you want to quit the DB Server Utility until your DBA has had an opportunity to create the Siebel Schema, click No. You can relaunch the wizard at any time to continue with this step, by navigating to `$SIEBEL_ROOT/bin` and entering:

```
srvrupgwiz /m master_install_mf.ucf
```

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

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

The following files are generated by the installation into the default output directory (dbsrvr/db2390/dboutput/install) or into the output directory that you designated in Step 17 on page 143.

- **schema.sql.** Contains DDL for all Siebel DB2 objects. You will transfer this file to the mainframe.
- **ddlsplit.txt.** Contains a REXX routine that can be used to split the DDL into more manageable pieces. This file must be moved to a PDS with RECFM = 80 and a LRECL of 80. You will transfer this file to the mainframe.
- **ddlview.sql.** Contains DDL for all Siebel views. You will transfer this file to the mainframe.
- **instddl.txt.** You will transfer this file to the mainframe. Contains sample JCL to execute the DDL using DSNTEP2. This file must be moved to a PDS with RECFM = 80 and a LRECL of 80. You will transfer this file to the mainframe.
- **instftp.txt.** Use this file to FTP files to the mainframe.
- **schmvld.sql.** Schema validation file.
- **seedvld.sql.** Seed validation file containing update statements that gather actual row counts for seed tables.

Applying DDL on the DB2 Host

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

Step 1: Transferring Files to the DB2 Host

Follow either one of the procedures below to preallocate space on the mainframe for the schema.sql and ddlview.sql files. This is a crucial step due to the size of the schema.sql file.

Method 1: Preallocating Space on DB2 Host using FTP Commands

The following FTP commands can be used to preallocate space. These commands are issued in the instftp.txt file transfer protocol (FTP) script that was installed in the output directory that you designated in Step 14 on page 134 (for Windows) or Step 17 on page 143 (for UNIX).

To preallocate space on DB2 Host using FTP commands

- Enter the following FTP commands:

```
quote site cylinders primary=25 secondary=2

quote site recfm=fb lrecl=80 blksize=0

put schema.sql
```

Method 2: Preallocating Space Directly on DB2 Host

If you do not use the FTP script (instftp.txt) to transfer the files to the mainframe, you can use the following procedure to preallocate space on the DB2 Host.

To preallocate space directly on DB2 Host

1. Access the Data Set Allocation panel, using ISPF option 3.2.
2. From the Data Set Allocation panel, allocate `CYLINDERS` as the space unit.
3. Allocate a quantity corresponding to the size of the schema.sql file (estimated at 25 cylinders).
4. For Record Format, select `FB`.

5. For record length, select `80`.

NOTE: If you will use SPUFI to execute the entire schema.sql DDL file (without splitting it into smaller files using the DDLSPLIT rexx exec), you must verify that the SPUFI output data set is allocated approximately three times more space than the data set allocated for schema.sql.

Step 2: Splitting the DDL

To split the DDL into multiple members in a PDS, you will use REXX DDLSPLIT (filename ddlsplit.txt). The input files for REXX DDLSPLIT are the DDL files generated by the installation process. The output from ddlsplit are multiple members in a PDS. DDLSPLIT creates the following members:

- DB - Create database statements.
- TS - Create table-space statements.
- TB - Create table and index statements.
- GRANT - All grant statements.

The DDLSPLIT rexx exec must be in a data set allocated to `sysproc` or `sysexec`.

When you are ready to execute DDLSPLIT, execute the following command on the TSO Command Processor panel (option 6 on most site installations):

```
TSO DDLSPLIT
```

If the REXX is not in a dataset allocated to `sysproc` or `sysexec`, then the entire dataset name must be specified; for example:

```
TSO EXEC 'SIEBEL.EXEC(DDLSPLIT)'
```

DDLSPLIT will prompt for the input dataset and the commit frequency. It will remove the existing commit statement, and it will commit at the specified frequency. If no frequency is specified, it will default to one (1). The DDLSPLIT rexx exec process dynamically allocates the output partitioned dataset.

After you create the storage control file, you are ready to run the database installation to create the DDL.

Step 3: Executing the SQL to Create the Siebel Schema

The DBA must now execute schema.sql and ddlview.sql on the DB2 host. You can use any DB2 tool (such as SPUFI or DSNTEP2) that you prefer to do this. It is recommended that you run in batch so that you do not restrict your TSO session. After the transfer, the DBA can edit the files, as desired, and then execute them, using any customary method.

You can use DDLSPLIT and REXX exec to split the schema.sql DDL file into manageable pieces for execution on the host. In order to use the DDLSPLIT utility, you need to copy the file named ddlsplit.txt from the dbsrvr/DB2390/DBOUTPUT/INSTALL directory to the mainframe.

Step 4: Setting Up the SQL Delimiter on DB2

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

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

```
CREATE TRIGGER Q202999.PTH0393 NO CASCADE BEFORE INSERT ON

Q202999.S_ACT_EMP REFERENCING NEW AS N FOR EACH ROW MODE DB2SQL

WHEN (N.EMP_ID IS NOT NULL) BEGIN ATOMIC SET N.PARTITION_COLUMN =

RIGHT(N.EMP_ID, 2); END

/
```

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

DSNTEP2

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

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

DSNTEP2 example:

```
//SYSTSIN  DD *

DSN SYSTEM(Q202)

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

       LIB('DSN710.RUNLIB.LOAD')

END
```

SPUFI

If you will use SPUFI, perform the following procedure.

To set the delimiter in SPUFI

- Set CHANGE DEFAULTS to YES.

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

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

To replace slashes with semicolons in the TSO editor

- Enter the following command:

  ```
  CHANGE X’61’ X’5E’ ALL
  ```

If your platform at midtier is Microsoft Windows, return to Step 25 on page 137; if your platform is UNIX, Step 30 on page 146.

Validating Siebel Schema Creation

When you manually transfer DDL to the mainframe (using the installation option Generate DDL into Files), you can validate created tables to make sure that no required tables were overlooked. To validate the tables, run the validation program after you install the Siebel Schema and before you install the seed data.

The validation consists of two parts:

- Creating a temporary table to hold the Siebel-provided logical schema
- Querying to identify any missing tables

Creating a Temporary Table

To create the temporary table TMP_S_SEBL_SCHM in a temporary EBCDIC database, you must have DBADM/CREATDBA privileges. The temporary table for validation is created in its own database and table space. The temporary database is created in EBCDIC format because of the requirement to do joins with the DB2 catalog.

The task takes approximately 30 minutes to complete. Make sure that you perform this task before you install seed data, as described above.

Siebel eBusiness Applications deliver the following number of tables and indexes for each product line:

- Horizontal—2288 tables and 11,656 indexes. After you complete this procedure, TMP_S_SEBL_SCHM should contain 13,944 records.
- Siebel Industry Applications (SIA)—3411 tables and 20,727 indexes. After you complete this procedure, TMP_S_SEBL_SCHM should contain 24,138 records.

To create a temporary table on OS/390

1. Locate the file SCHMVLD within the *`DBSERVER_ROOT`* subdirectory.
2. If you prefer to edit the file on the host, transfer the file to OS/390.
3. In order to run the validation scripts, input the names for the following parameters:
   - *`CREATOR`* = The name of your newly installed schema qualifier.

- *TMPDB* = The name of the temporary database.
- *SCHMTS* = The name of the table space.
- *DEFLTSTO* = The name of the default storage group.

Once the values have been entered for the above parameters, the header and comments in the SCHMVLD file must be removed. The header and comments are not compliant with DB2 format.

4 Submit the SQL by using either a native DB2 tool (if working on the DB2 host) or by using one of the DB2 Client tools, such as the DB2 Command Window or the DB2 Command Line Processor (if working from a Siebel Server).

During the script execution, the error message `SQL return code -204` could be encountered when performing `DROP TABLE` *`CREATOR`*`.TMP_S_SEBL_SCHM`. Ignore this message. It is safe to drop the temporary database after you complete the Siebel Schema validation.

Querying to Identify Missing Tables

You must issue the following query to identify missing tables either before or after the following installation prompt appears, but before you select `Yes`.

```
Please create Siebel Views using ddlview.sql located in the DDL
Output Directory. Once the views are created, please select Yes. To
stop now and continue installation later please select No. (To resume
at a later time, please start the Upgrade Wizard from the command
line using option /m master_install_mf.ucf)
```

To identify missing tables

1 From the DB2 Command Window or the DB2 Command Line Processor, issue a query to identify missing tables. Examples of queries to identify missing tables are listed below. Edit these queries as necessary.

```
SELECT NAME “MISSING TABLE” FROM SCHEMA_QUALIFIER.TMP_S_SEBL_SCHM
WHERE TYPE = ‘T’
AND NAME NOT LIKE ‘V_%’
AND NAME NOT LIKE ‘W_%’
AND NAME NOT IN (SELECT NAME FROM SYSIBM.SYSTABLES
WHERE CREATOR = ‘SCHEMA_QUALIFIER’
);
```

where:

SCHEMA_QUALIFIER = the schema qualifier for the installation

2. Review the output for missing tables and reapply the corresponding DDL as necessary.

Preparing the Database Layout

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

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

- **Use an existing storage control file.**
- **Modify one of the Siebel-provided storage control file templates.** To modify a template, you can edit the file directly, or you can use a Siebel tool called the Database Storage Configurator (dbconf.xls). For more information, see "About Creating or Modifying the Database Layout" on page 80
- **Extract a new storage control file for custom installation.** To create a storage control file based on the layout of the existing Siebel Schema, use the DB Server Configuration Utility—Extract Utility. This utility allows you to create the storage control file by "reverse-engineering" the DB2 catalog. For more information, see "Extracting a Storage Control File from the DB2 Catalog."

Extracting a Storage Control File from the DB2 Catalog

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

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

NOTE: You must validate the storage control file after you extract it and after you modify it. You must also perform a validation before you apply the schema.sql on the DB2 host. See "Validating the Storage Control File" on page 161.

Accessing the Utility Under Windows

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

To extract the storage control file under Windows

1. Launch the Database Server Configuration Utility:

   Choose Start > Programs > Siebel Enterprise Server 7.x.x > Configure DB Server.

   or

   a Open a command prompt.

   b Go to the `SIEBSRVR_ROOT\bin` directory.

   c Run the following command:

   ```
   ssincfgw -l ENU -v Y
   ```

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

2. In the Gateway Server Address screen, type the following values as you recorded them in your copy of Appendix A, “Deployment Planning Worksheet.”

   Gateway Server Address. The alias of the host on which you installed the Siebel Gateway.

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

   To continue, click Next.

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

   - Accept the default value displayed in the Siebel Server Directory field. (This is the *SIEBSRVR_ROOT* directory, for example, `D:\sea7xx\siebsrvr`.)
   - Use the browse button to select an alternate directory path.

   To continue, click Next.

4. In the Siebel Database Server Directory screen, perform either of the following tasks:

   - Accept the default path displayed in the Siebel Database Server Directory field. (This is the *`DBSRVR_ROOT`* directory, for example, `D:\sea7xx\dbsrvr`.)
   - Use the Browse button to select a different database directory.

   To continue, click Next.

5. In the RDBMS Platform screen, choose `IBM DB2 UDB for OS/390 and z/OS`, and then click Next.

6. In the Siebel Database Operation screen, select `Run Database Utilities`, and then click Next.

7. In the Database Utility Selection screen, select `Configure Database`, and then click Next.

   The Configure Database screen appears with the following options:

   - Extract Storage Control File.
   - Validate Storage Control File. This will validate that a specified template and a specified schema will work on the DB2 host.

8. In the Database Configuration Options screen, select `Extract Storage File`, and then click Next.

9. In the OBDC Data Source Name screen, accept the default name of the ODBC data source after verifying that it is correct (make sure that this data source has the same name as the subsystem qualifier), or enter a different data source name.

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

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

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

11 In the Siebel Schema Qualifier screen, type the schema qualifier and group ID for your existing database, and then click Next.

The schema qualifier is also an authorization ID. The schema qualifier must start with a letter and cannot contain special characters.

12 In the Security Group ID screen, type the user ID of the group to whom schema access is granted, for example, `SSEROLE`.

To continue, click Next.

13 In the Extract Options screen, select `Extract from Catalog`, and then click Next.

This option extracts the storage layout of your source database from the DB2 catalog as-is. The output will be a new storage layout.

14 In the Storage File screen, type the name of the new storage control file, for example *`DBSRVR_ROOT`*`\DB2390\`*`my_storage_file.ctl`*, and then click Next.

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

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

- If you need to go back to make changes, click Previous to back out until you reach the screen with the parameter you need to change. Enter the valid parameter, and then click Next until you reach the Configuration Parameter Review screen again.

- To accept the values you input with no changes, click Finish. A message box gives you the choice of applying the configuration now or applying it later. This screen also provides the option you will need in order to apply the configuration later from the command line.

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

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

The Siebel Upgrade Wizard creates the storage control file.

Accessing the Utility Under UNIX

Follow the procedure below to access the utility under UNIX.

To create the storage control file under UNIX

1 Source environment variables from `$SIEBEL_ROOT`.

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

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

NOTE: *`LANGUAGE`* is not the same as *`LANG`*; the former is the language in which the configuration wizard runs, while the latter is the code for the language pack you installed.

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

csh

```
setenv SIEBEL_ROOT New Value

setenv LANGUAGE New Value
```

ksh

```
SIEBEL_ROOT=New Value ; export SIEBEL_ROOT

LANGUAGE=New Value; export LANGUAGE
```

2. Go to $*SIEBEL_ROOT* /bin and enter:

   ```
   dbsrvr_config.ksh
   ```

   This launches the Database Server Configuration Wizard.

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

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

   ---

4. Specify the path of your Siebel Server root directory, or accept the default by pressing ENTER.

5. Specify the path of your database server root directory, or accept the default by pressing ENTER.

6. Enter the number that corresponds to your database platform.

7. In the Select Database Operation screen, choose Run Database Utilities (5).

8. In the Database Utility Selection screen, choose Configure Database (3).

9. In the Database Configuration Options screen, choose Extract Storage Control File (1).

10. Enter the number that corresponds to the language you are using, typically `1` for English.

11. Enter your source ODBC connection name.

12. Enter your source database username and password, and then confirm the password by typing it again.

13. Enter the schema qualifier in uppercase letters for your existing database.

14. Enter the group ID for your existing database, or accept the default (SSEROLE) by pressing ENTER.

15. In the Extract Options screen, select Extract from Catalog (1).

16. Enter the name of the new storage control file, for example `DBSRVR_ROOT/DB2390/my_storage_file.ctl`

17. When the configuration parameters are listed for your review and you are asked if you want to run `srvrupgwiz`, enter Y for yes.

18. When the Siebel Upgrade Wizard launches, enter Y to perform the operation.

    The Siebel Upgrade Wizard creates the storage control file.

Validating the Storage Control File

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

- `db_config_siebel.log`
- `db_config_validation.log`

To validate the storage control file under Windows

1. Run the Siebel Software Configuration Utility as you did in "Extracting a Storage Control File from the DB2 Catalog" on page 155.

2. From the Database Utility Selection screen, choose `Configure Database` and click Next.

   The Configure Database screen appears with the following options:

   - Extract Storage Control File
   - Validate Storage Control File

3. Choose `Validate Storage Control File` and click Next.

4. On the ODBC Data Source screen, accept the default name of the ODBC data source after verifying that it is correct (make sure that this has the same name as the subsystem qualifier), or enter a different data source name.

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

   To continue, click Next.

5. On the Database User Name screen, indicate the source user name and password for your database:

   Database User Name. Type the ID allowed to log in to the Siebel Database. This user ID should have authorization to set CURRENT SQLID.

   Database Password. Type the password for the ID used to log in to the Siebel database.

   Database Password (confirm). Retype the password to confirm it.

   To continue, click Next.

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

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

   To continue, click Next.

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

   To continue, click Next.

8 On the Schema File screen, type the directory paths for your schema file (for example, schema.ddl) and storage control file (for example, storage.ctl).

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

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

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

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

A message box appears with the prompt:

```
To apply the configuration now, press “OK”.

To apply the configuration later, press “Cancel”.
```

10 To validate that storage.ctl works, connect to catalog, run ddlimp in validation mode.

11 Review the log file (`ddlimp.log`) that is generated in the log subdirectory under *SIEBELSRVR_ROOT*.

To validate the storage control file under UNIX

1 Verify that all servers are stopped:

- Stop the Gateway Name Server by typing: `stop_ns`
- Stop the Siebel Servers by typing: `stop_server ALL`

2 Depending on your shell, source environment variables from *$SIEBEL_ROOT* by typing:

   Korn shell

   ```
   . siebenv.sh
   ```

   C shell

   ```
   source siebenv.csh
   ```

3 Navigate to $*SIEBEL_ROOT* /bin and enter:

   ```
   dbsrvr_config.ksh
   ```

   This launches the Database Server Configuration Wizard.

4 Review the values of the following environment variables and confirm whether or not the settings are correct by entering either Y or N.

   - SIEBEL_ROOT
   - LANGUAGE

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

5 Specify the path of your Siebel Server root directory or accept the default by pressing Enter.

6 Specify the path of your database server root directory or accept the default by pressing Enter.

7 Enter the number that corresponds to IBM DB2 UDB for OS/390 and z/OS, then press Enter.

8 In the Select Database Operation screen, choose Run Database Utilities (5).

9 In the Database Utility Selection screen, choose Configure Database (3).

10 In the Database Configuration Options screen, choose Validate Storage Control File (2).

11 Enter your source ODBC connection name.

NOTE: This needs to be the same as the actual name of your subsystem. That is, your database alias within DB2 Connect must be the same as the actual name of your DB2 subsystem.

---

12 Enter your source database username and password.

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

14 Enter the security group ID for your existing database, or accept the default (SSEROLE) by pressing Enter.

15 Specify the storage control file that you would like to validate.

The configuration parameters are listed.

16 When the program asks if you would like to run `srvrupgwiz`, enter Y for Yes.

The storage control file is validated.

17 Review the log files generated by ddlimp (`ddlimp.log`) that is generated in the log directory under *`SIEBELSRVR_ROOT`*.

Using the Siebel Database Storage Configurator

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

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

After you have edited a storage control file template using the Database Storage Configurator, save the file with a new name, then direct the DB Server Configuration Utility to your newly created (saved) and configured storage control file during installation (as described in "Performing a Custom Installation" on page 176). When the Database Server Configuration Utility prompts you to specify a storage control file, select your new storage control file. The output DDL is generated from the information in the storage control file that you select.

Configuring a Siebel Storage Control File

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

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

To configure one of the Siebel storage control files

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

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

When the spreadsheet opens, the Home tab is active.

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

a Click Import.

b Go to the directory where your files with a `.ctl` suffix are located.

c Double-click the appropriate file.

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

Click OK.

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

> **NOTE:** Replace variables that are preceded with “$” with actual values for your deployment.

- Table Storage Group for Tables
- Index Storage for Indexes
- 4-KB Buffer Pool Name
- 16-KB Buffer Pool Name
- 32-KB Buffer Pool Name
- Index Buffer Pool Name

- Database Name Prefix
- Encoding Scheme (either ASCII- or EBCDIC-based)

4. Click Set.
5. Display the template defaults by clicking the Default tab.
6. Review the values and update them as required.

   NOTE: The default values are the combined values that apply to each individual object represented by the tabs; the values can be changed within each of those tabs.

   The parameters in the spreadsheet correspond to native DB2 syntax. For example, the parameters `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 (SIEBS00) values of the template used unless you explicitly define new values.

7. Review the databases by clicking the Database tab.

   Be aware that the values visible on any tab also reflect the defaults for any objects at a lower level from the object for which the tab exists. So, if no locksize is shown on for a table space under the Tablespace tab, that means that it takes the default from the database object it belongs to; for example, locksize on the database object level applies to all table spaces created in a given database unless the locksize is specified explicitly in the table space object.

8. Review the default table spaces by clicking the Tablespace tab.

   To review any partitions, do either one of the following:

   - Click twice on the Tablespace tab. (Clicking on the Tablespace tab alternates between the table spaces and table space partition worksheets.)
   - Select a row that displays a value in the Partition column greater than zero, then click Show TSPARTs.

   This displays the rows of corresponding partitions.

9. Edit the values as desired, step off the cell to save the value.

10. When you are done updating the template, click the Home tab and then click Export.

    The Export worksheets to storage control file screen appears.

11. Rename the file and save it to a location that is easy for you to find during installation.

12. To make the Configurator ready for use with a different template, from the Home tab, click Clean.

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

Validating Your Work

You can use the Validation button at any time to validate the syntax in the spreadsheet. Your worksheets will be more consistent if you wait to validate until after you have finished editing.

To validate your worksheet

1. Click the Validate button.

   The Storage Configurator validates the new value and if the value is incorrect it appears in red. (A text box also appears prompting you to review the invalid settings in red text.)

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

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

Partitioning Tables and Indexes

The following two procedures demonstrate how to use the Database Storage Configurator to partition tables after you have worked out a partitioning scheme. For guidelines about partitioning, see Chapter 4, “Understanding Siebel Partitioning.”

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

Decreasing the Number of Partitions

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

To decrease the number of partitions

1 Click the Tablespace tab on the Database Storage Configurator, which should still be open from the previous procedure, “Configuring a Siebel Storage Control File” on page 166.

NOTE: If you need to relaunch the Database Storage Configurator, you first must import the .ctl file.

2 Locate the row with the Tablespace tab in it that you want to change, and change the value in the Partitions column to reflect the number of new partitions you want to create.

3 Save the new value by stepping off the cell.

4 Click Show TSPARTs.

- If a message box appears, prompting you that there are additional table space partitions for this table, and requesting that you review and delete them, do so now.
- If everything is in order, you receive a message indicating that the operation was successful.

5 Click the Partitioning Index tab and locate the partitioning index associated with this table space.

6 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.

7 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 8.
- If everything is in order, you receive a message indicating that the operation was successful.

8 Review the values in the LimitKey column and update them appropriately.

NOTE: When using Excel, you must type the first quote as two single quotes. Excel saves it as a single quote when you step off the cell. If you do not enter two single quotes, Excel deletes it entirely when you step off the cell.

9 Make any other edits needed to the template you are using and follow the steps under "Configuring a Siebel Storage Control File" on page 166 to verify and save your changes to a location that will be easy to find when you or someone else installs the Siebel Schema.

Increasing the Number of Partitions

Usually, the partitioning is done for a table. You must adjust the related table space and partitioning index. The following procedure describes how to increase the number of partitions for a table and index in a given table space.

To increase the number of partitions

1. Launch the Database Storage Configurator as described earlier in this chapter under "Configuring a Siebel Storage Control File" on page 166 and select the Table tab.
2. Locate the name of the partitioned table space for which you want to increase the number of partitions.
3. Click the Tablespace tab and search for the corresponding table space.
4. Increase the value in the Partitions column for that table space as appropriate and step off the cell to save it, but keep your cursor in the same row. (If you do not keep your cursor in the same row, you cannot get to the table space partitions by clicking the Show TSPARTs button.)
5. Click Show TSPARTs.

   When you increase the value in the Partitions column, you receive a message indicating that the partitions were added successfully.
6. Click the Partitioning Index tab and locate the partitioning index associated with the table you are operating on.
7. Increase the number of partitions to match the value you previously entered for the associated table space, step off the cell to save the value, but keep your cursor on the same row.
8. Click Show Index Parts.

   A text box appears prompting you that the extra partitions were created successfully.
9. Review the LimitKey values and update them as appropriate.

   NOTE: When using Excel, you must type the first quote as two single quotes. Excel saves it as a single quote when you step off the cell. If you do not enter two single quotes, Excel deletes it entirely when you step off the cell.

10 Make any other edits needed to the template you are using. Then follow the steps under "Configuring a Siebel Storage Control File" on page 166 to verify and save your changes to a location that will be easy to find when you or someone else installs the Siebel Schema.

Working with CLOBs

Your Siebel application is delivered with the objects needed to create character large objects (CLOBs), even if you did not choose CLOBs as an option. These objects are used to create CLOBs only at the table level to prevent excessive processing overhead.

To activate a CLOB

1 Launch the Database Storage Configurator as described earlier in this chapter under "Configuring a Siebel Storage Control File" on page 166 and select the Table tab.

2 Review the contents of the CLOB column.

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

NOTE: The auxiliary table space name displays a table name that is used as a pointer to a real auxiliary table space. The parameters displayed under this tab are the only parameters allowed to change for the auxiliary object.

Creating and Executing the DDL

You are ready to transfer the output DDL (generated from the information in the storage control file that you selected) to the mainframe, where you will execute them.

Moving DDL Files to the S/390 or zSeries Machine

When you are ready to transfer DDL files for execution, you will use FTP (or any other available file transfer option) to move the following files from the output directory (the default output directory is dbsrvr\db2390\dboutput\install on Windows or dbsrvr/db2390/dboutput/install on UNIX) to the S/390 or zSeries machine:

- **Ddlsplit.txt.** Contains a REXX routine that can be used to split the DDL into more manageable pieces. This file must be moved to a PDS with RECFM = 80 and a LRECL of 80.
- **Instddl.txt.** Contains sample JCL to execute the DDL using DSNTEP2. This file must be moved to a PDS with RECFM = 80 and a LRECL of 80.
- **Schema.sql.** Contains DDL for all Siebel DB2 objects.
- **Alter statements.** (Optional) Contains the alter statements for the object sizing. This file must be moved manually.

A sample FTP file named instftp.txt is shipped with Siebel eBusiness Applications.

Splitting the DDL

To split the DDL into multiple members in a PDS, you will use Rexx DDLSPLIT (filename ddlsplit.txt). The input files for REXX DDLSPLIT are the DDL files generated by the installation process. The output from ddlsplit are multiple members in a PDS.

DDLSPLIT creates the following members:

- DB - Create database statements.
- TS - Create table-space statements.

- TB - Create table and index statements.
- GRANT - All grant statements.

When you are ready to execute DDLSPLIT, use the following command:

```
TSO DDLSPLIT
```

If the REXX is not in a dataset allocated to `sysproc` or `sysexec`, then the entire dataset name must be specified; for example:

```
TSO EXEC 'SIEBEL.EXEC(DDLSPLIT)'
```

DDLSPLIT will prompt for the input dataset and the commit frequency. (The default commit frequency is 10.) It will remove the existing commit statement, and it will commit at the specified frequency. If no frequency is specified, it will default to one (1). The DDLSPLIT rexx exec process dynamically allocates the output partitioned dataset.

After you create the storage control file, you are ready to run the database installation to create the DDL.

Executing Alter Statements

After you have executed all the DDL, you will execute the alter statements to change the object sizes. This must be done prior to inserting any data into the tables. All objects have been defined with DEFINE NO. This means that the alter statements will be used at first insert when the physical datasets are created. If data has been inserted then the alter statements will not change the actual size until after a DB2 reorg or load.

Once all objects have been created, the database installation can be resumed on the midtier.

Performing a Custom Installation

The custom installation option provides significant flexibility for configuring the storage layout for the Siebel application installation.

The custom installation consists of the following steps:

- Prepare a storage control file. (See "Preparing the Database Layout" on page 155.)
- Run the Database Server Configuration Utility, select the Custom Installation option, and provide your customized storage control file (prepared during the previous step) that will be used in the storage configuration for the Siebel Schema. (See the following procedure for your operating system.)
- Populate the seed data. (See the following procedure for your operating system.)
- Import the Siebel Repository. (See Chapter 7, "Siebel Repository Import, Export, and Postinstallation Tasks.")

Performing a Custom Installation Under Windows

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

To perform a custom installation under Windows

1. Launch the Database Server Configuration Utility and proceed as in the Standard Installation option until the Select Installation Type screen appears (Step 12 on page 132).
2. In the Select Installation Type screen, select Customized Installation.

   To continue, click Next.

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

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

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

  You must later use FTP or a similar file transfer program to transport the DDL (schema.sql and ddlview.sql) to the DB2 host, where the DBA executes it, using customary methods, for example, SPUFI.

  To continue, click Next and proceed to Step 4.

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

  The Siebel application uses ODBC to apply all necessary steps in this installation.

  To continue, click Next and proceed to Step 5 on page 178.

4 In the DDL Output Directory screen, indicate the following values.

- **DDL Output Directory.** Accept the default directory or select another location (by typing the location or by using the Browse button) where you want to deposit the generated DDL for later transport to the DB2 host. The default directory uses the following syntax:

  `SIEBEL_ROOT\DBSRVR\DB2390\DBOUTPUT\INSTALL`

- **Storage Control File.** Use the Browse button to indicate the path to the customized storage control file that you want to use to configure storage on the DB2 host.

  This can be either a Siebel-provided template or a storage control file that you have already customized using any of the methods in "Preparing the Database Layout" on page 155.

  To continue, click Next and proceed to Step 6 on page 178.

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

   This is the storage control file that you have customized already using any of the previously described methods.

   To continue, click Next.

6. In the Configuration Parameter Review screen, review the configuration values you entered on the previous Configuration Utility screens:

   - To go back and make changes, click Previous to back out until you reach the screen with the parameter you need to change.

   - To accept the values you input with no changes, click Finish.

     The following message appears:

     ```
     To apply the configuration now, press "OK".
     To apply the configuration later, press "Cancel".
     The command line to apply the configuration later is
     C:\sea7xx\siebsrvr\bin\siebupg /m master_install_mf.ucf
     ```

     NOTE: The last line of the prompt is generated dynamically and might vary based on the location of your Siebel Server installation.

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

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

7. When the Database Server Configuration Utility displays the items to be executed or imported, click OK to start generating DDL.

   - This step completes the installation option Run DDL Automatically.

   - If in Step 3 on page 173 you chose the installation option Generate DDL Into Files, the utility displays the following message after it generates the schema.sql file that contains the DDL to create the Siebel Schema:

     ```
     Please create Siebel Schema using schema.sql and ddlview.sql
     located in the DDL Output Directory. Once the views are
     created, please select Yes. To stop now and continue
     installation later please select No. (To resume at a later
     time, please start the Upgrade Wizard from the command line
     using option /m master_install_mf.ucf).
     ```

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

   - When you are prompted to create the Siebel Schema and views, click Yes to continue.

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

   For DBA information on how to apply the Siebel Schema on the DB2 host, see "Applying DDL on the DB2 Host" on page 148.

9. Indicate which action you want to take:

   - If your DBA has already created the Siebel Schema, click Yes to apply DML to populate it. (This action does not require DBA authority.)

     CAUTION: When you click Yes, it means you want to populate the schema with seed data, among other things. The operation will fail if the Siebel Schema has not yet been created, because there is no schema to populate.

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

When you have successfully applied the seed data to the Siebel Schema, you have finished installing the Siebel Database Server components.

TIP: Locate the file master_install_mf.ucf in the binary subdirectory of your Siebel Server installation directory and print it out, so that you have a record of the values you entered.

After you create the storage control file, you can run the database installation to create the DDL.

Reviewing the Log Files for Windows

The database server component installation creates a number of log files, such as UpgWiz.log, UpgWiz_01.log (the numeric suffix incrementing for subsequent log files), within the log subdirectory of the siebsrvr installation directory. Review the log files for any errors.

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

The main log file, produced by the Siebel Upgrade Wizard (UpgWiz.log), does not contain errors returned by applications. Additional information about the configuration process can be found in the sw_cfg_utl.log file. Depending on the operation you performed, the ddl_ctl.log file reflects any errors.

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

```
GenericLog GenericError 1 2002-02-28 15:51:24 (err=1) was
returned by application (D:\sea704\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 one above, the ddlimp.log file contains a detailed error message:

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

A Siebel utility called htmlgenm can also be helpful by creating an HTML file with the summary report of all steps performed by the Upgrade Wizard together with the information on log files.

Three options exist for the use of htmlgenm:

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

For example, if you want to see the HTML results from the database installation, the following command generates the summary.htm file in the log subdirectory:

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

Performing a Custom Installation Under UNIX

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

To perform a custom installation under UNIX

1. Follow the steps previously documented for performing a standard installation on the UNIX operating system. Continue until you reach Step 15 on page 142, and enter `2` for `Customized Install`.

2. Enter the number that corresponds to the mechanism you want to use for installation:

   - `1 - Generate DDL Into Files`. Generates DDL required to create the Siebel Schema. Proceed to Step 3.

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

     You must later use 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.

   - `2 - Run DDL automatically`. Select this to apply the DDL and DML required to create the Siebel Schema directly against the database. The Siebel application uses ODBC to apply the required DDL to the DB2 host. Proceed to Step 4 on page 183.

3. Enter the path to the output directory for the DDL files you will transfer to your zSeries machine, or accept the default directory:

   ```
   dbsrvr/db2390/dboutput/install
   ```

   NOTE: If you plan to use a different output directory to hold the DDL than the Siebel default, you must create it before selecting this option.

4. Enter the name of the customized storage control file that you created and saved earlier; this file will be used to configure storage on DB2, including the full path to its location.

   This can be either a Siebel-provided template or a storage control file that you have already customized using any of the methods in "Preparing the Database Layout" on page 155.

5. Review the list of the parameters you input. If they are correct, enter `Y` to generate the Siebel Schema.

   If `Run DDL automatically` is chosen, the Siebel Schema is applied directly to DB2. If `Generate DDL into Files is` chosen, the `schema.sql` and `ddlview.sql` files are created.

Reviewing the Log Files for UNIX

The database server component installation creates a number of log files, such as srvrupgwiz.log, srvrupgwiz1.log (which increments to additional logs) within the `log` subdirectory of the `siebsrvr` installation directory, which you must review for any errors.

When you install your base language pack, it creates a special log file with the name dataimp_prim_lang.log.

The main log file, produced by the Siebel Upgrade Wizard (Srvrupgwiz.log), does not contain errors returned by applications. Additional information about the configuration process can be found in the sw_cfg_utl.log file. Depending on the operation performed, ddl_ctl.log file reflects any errors:

To find additional operation-specific log files, open Srvrupgwiz.log, using any text editor, and search for a ".log" string.

This shows the names of log files, produced during the operation. For example, if the configuration of the Database Server fails, the Srvrupgwiz.log contains the following information:

```
GenericLog GenericError 1 2002-02-28 15:51:24 (err=1) was
returned by application (D:/sea704/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:

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

The log files might include errors that are expected and benign. Compare any error messages found in the log files to the sample error messages in the errors.rtf file that is located in the db2390 subdirectory of your `dbsrvr` installation directory. (If a log file is not listed in the errors.rtf file, then there are no acceptable error messages for that log file.) No further action is required if the log files contain only errors listed in the errors.rtf file.

NOTE: Only one of each type of error occurring in a particular log file appears in the errors.rtf file.

If you find errors not listed in the errors.rtf file, correct the condition that caused the errors, and rerun the Upgrade Wizard. The wizard restarts from the point where it left off.

Do not review error numbers alone, because error numbers might change after installation of a new driver version. Instead, compare the actual error descriptions to find out which are acceptable errors for this platform.

CAUTION: Although other errors are rarely encountered, this review is critical. Certain errors, such as a failure to create indexes, might result in performance problems or anomalous behavior in Siebel eBusiness Applications.

Rerunning the Installation

If you need to rerun the installation from the beginning, either after completing an installation run or after starting but not finishing the run, delete the driver_install_mf subdirectory from the upgrade subdirectory of your `siebsrvr` installation directory.

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

Siebel Repository Import, Export, and Postinstallation Tasks 7

This chapter describes how to import and export the Siebel Repository, and provides both mandatory and optional postinstallation procedures. For an overview of the involved tasks, see Table 15.

Table 15. Database Server Installation and Configuration Tasks

Who Performs It?	Task
Siebel administrator	**1** Import the Siebel Repository. See "Importing and Exporting the Siebel Repository" on page 188.
	2 Review the repository import log. See "Reviewing the Log Files for Repository Import" on page 197.
	3 Perform any necessary troubleshooting. See "Troubleshooting the Repository Import" on page 198.
Database administrator (DBA)	**4** Grant table privileges and load statistics. See "Granting Table Privileges" on page 199.
Siebel administrator and DBA	**5** Run dbchck to validate the physical schema against the Siebel Repository. See "Validating the Siebel Schema" on page 200.
	6 If you are deploying multiple languages, install multilingual seed data. See "Installing Multilingual Seed Data" on page 202.
	7 If you are deploying multiple languages, import multilingual seed data to your repository table rows. See "Importing a New Language to Your Repository" on page 207.
DBA	**8** Populate the Siebel File System. See "Populating the Siebel File System" on page 211.

Importing and Exporting the Siebel Repository

Repository import is the final step in the Siebel Schema installation process.

When you import the Siebel Repository, you populate all the repository tables in the Siebel Database Server with Siebel eBusiness Applications, Release 7.0 application objects.

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

The Siebel Repository can also be exported into a platform-independent file that can be sent to Siebel Technical Support for analysis in case of problems that cannot be diagnosed by telephone or email.

To import the Siebel Repository under Windows

1. Launch the Database Server Configuration Utility, using any method described in Chapter 2: “Preparing for the Installation” of the *Siebel Server Installation Guide for Microsoft Windows*.

2. In the Gateway Server Address screen, type the following values as you recorded them in your copy of Appendix A, “Deployment Planning Worksheet.”

   - **Gateway Server Address.** The alias of the host on which you installed the Siebel Gateway.

   - **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 Server Address and Enterprise Server Name fields blank.

3. In the Siebel Server Directory screen, perform either of the following tasks:

   - Accept the default value displayed in the Siebel Server Directory field. (This is the *`SIEBSRVR_ROOT`* directory, for example, `D:\sea7xx\siebsrvr`.)

- Use the browse button to select an alternate directory path.

To continue, click Next.

4. In the Siebel Database Server Directory screen, perform either of the following tasks:

   - Accept the default path displayed in the Siebel Database Server Directory field. (This is the *`DBSRVR_ROOT`* directory, for example, `D:\sea7xx\dbsrvr`.)

   - Use the Browse button to select an alternate directory.

   To continue, click Next.

5. In the Database Platform screen, select the database platform you use, in this case, DB2 UDB for OS/390 and z/OS.

   To continue, click Next.

6. In the Siebel Database Operation screen, select Import/Export Repository, and then click Next.

7. In the Select Repository Operation screen, choose 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 8 on page 190.

   - **Add Language to an Existing Repository.** If you already imported your Siebel Repository and its base language, select this option to add another language to the repository. See “Importing a New Language to Your Repository” on page 207.

   NOTE: You must have already imported your base-language repository before you can add a new language.

- **Export Repository.** Exports the Siebel Repository into a platform-independent file that can be sent to Siebel Technical Support for analysis if needed. Proceed to Step 9 on page 190.

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

  To continue, click Next.

8. In the Import Selection screen, indicate the following:

   - To import a standard Siebel 7 repository, click the uppermost radio button.
   - To import a Siebel 7 repository that you have customized to a new environment, click the bottom radio button.

     NOTE: Select Import Custom Siebel 7 Repository when you are importing a multilingual repository from a test or development environment. This imports *all* languages to your target repository.

   To continue, click Next.

9. In the ODBC Data Source Name screen, enter the name of the ODBC data source, for example, `SiebSrvr_Siebel`, or enter the database alias you prefer to use for the data source. This alias should have been previously registered within DB2 Connect.

   NOTE: If you want to work with information on a different database server than the one you currently point to, you can enter the ODBC name of that database server, as long as you have previously registered this as an ODBC datasource within DB2 Connect.

   The Siebel Server installation process automatically creates the data source, using the syntax: `SiebSrvr_`*`enterprise_server_name`*.

   To continue, click Next.

10. In the Database User Name screen, indicate the following:

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

To continue, click Next.

11 In the Siebel Schema Qualifier screen, type the up to eight-character ID that identifies the Siebel Schema owner. This ID must be in uppercase, must start with a letter, and cannot include any special characters.

This is also the SQL Authorization ID stored in the CURRENT SQLID special register.

- If you are importing your repository for the first time, the Import Repository Name screen appears. Proceed to Step 12.
- If you are *exporting* your repository, the Export Repository Name screen appears. Proceed to Step 13.

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

- **Import Repository Name.** Type the designated name for this Siebel Repository.
- **Repository File Name.** If you are importing your repository for the first time, this is the second field appearing on the screen. Accept the default installation path and file name for this repository or type another valid installation path.

To continue, click Next. Proceed to Step 14.

13 In the Export Repository Name screen, type the following values:

- **Export Repository Name.** Accept the default name (Siebel Repository) or type another valid name.
- **Repository File Name.** Accept the default installation path and file name for this repository or type another valid installation path.

To continue, click Next.

14 In the Configuration Parameter Review screen, review the configuration values that you entered on the previous configuration screens:

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

A message box appears with the prompt:

```
To apply the configuration now, press “OK”.
To apply the configuration later, press “Cancel”.
The command line to apply the configuration later is
C:\sea704\siebsrvr\bin\siebupg /m master_imprep.ucf
```

NOTE: The last line in the prompt is generated dynamically based on the location of your Siebel Server installation.

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

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

15 When the DB Server Configuration Utility displays the items to be executed or imported, click OK to begin.

A window appears, displaying information about repository import activities. Afterwards, a message appears when repository import is complete.

To verify that the import was successful, review the log files. See “Reviewing the Log Files for Repository Import” on page 197.

TIP: Locate the file master_imprep.ucf in the binary subdirectory of your Siebel Server installation directory and print it out, so that you have a record of the values you entered.

If you need to change the values you previously entered, relaunch the Configuration Utility by double-clicking the Configure DB Server icon within Start > Programs > Siebel Enterprise Servers 7.x.x. This allows you to reconfigure with different values.

To import the Siebel Repository under UNIX

1 Source environment variables from $*SIEBEL_ROOT*.

SIEBEL_ROOT should be the path of your Siebel installation directory.

LANGUAGE should be set to the language in which the Configuration Wizard prompts appear, for example, enu for U.S. English.

NOTE: *LANGUAGE* is not the same as *LANG*; the former is the language in which the Configuration Wizard runs, while the latter is the code for the Language Pack you installed.

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

csh

```
setenv SIEBEL_ROOT New Value
```

```
setenv LANGUAGE New Value
```

ksh

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

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

2 Go to $*SIEBEL_ROOT* /bin and enter: `dbsrvr_config.ksh`

This launches the Database Server Configuration Wizard.

3 Review the values of the following environment variables and confirm whether the settings are correct by entering either `Y` or `N`:

SIEBEL_ROOT = path of your Siebel installation directory

LANGUAGE = language in which you are installing

NOTE: If the values for either *SIEBEL_ROOT* or *LANGUAGE* are not set or are incorrect, you must correct them before proceeding.

---

4. Specify the path of your Siebel Server root directory, or accept the default by pressing ENTER.

5. Specify the path of your database server root directory, or accept the default by pressing ENTER.

6. Enter the number corresponding to your RDBMS platform, in this case, DB2 for OS/390 and z/OS `(2)`.

7. From the Siebel Database Operation menu, choose Import/Export Repository `(3)`.

8. Enter the number that corresponds to the procedure you want to complete:

   `1 - Import Repository.` To import the Siebel Repository for the first time with a base language, select this option. Proceed to Step 9 on page 195.

   `2 - Add Language to an Existing Repository.` If you already imported your Siebel Repository and its base language, select this option to add another language to the repository. See "Importing a New Language to Your Repository" on page 207.

   NOTE: You must have already imported your base-language repository before you can add a new language.

   `3 - Export Repository.` Exports the Siebel Repository into a platform-independent file that can be sent to Siebel Technical Support for analysis if needed. Proceed to Step 10 on page 195.

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

   If you selected Import Repository, the Import Selection screen appears.

9 Specify the applicable import option:

`1 - Import Siebel Repository.` To import one additional language to your repository, enter `1`. See "Importing a New Language to Your Repository" on page 207.

`2 - Import Custom Repository.` To import *all* languages to your repository, enter `2`. Proceed to Step 10.

NOTE: Option 2 is used when you are importing a multilingual repository from a test or development environment.

---

10 Enter the number corresponding with the language in which you want to run your database. (This is known as your base language.) If you are installing your first language, this is (`1`).

11 Enter the name of the ODBC data source, for example, `SiebSrvr_Siebel`, or enter the database alias you prefer to use for the data source. This alias should have been previously registered within DB2 Connect.

NOTE: If you want to work with information on a different database server than the one you currently point to, you can enter the ODBC name of that database server, as long as you have previously registered this as an ODBC datasource within DB2 Connect.

The Siebel Server installation process automatically creates the data source, using the syntax: `SiebSrvr_`*`enterprise_server_name`*.

12 Specify the Database User Name.

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

13 Enter the database password.

14 Specify the Schema Qualifier.

This is the up to eight-character identifier that designates the Siebel Schema. This is also an authorization ID. The Schema Qualifier must start with a letter and cannot contain special characters. It must also be in upper case.

15 Specify the group ID or accept the default (SSEROLE) by pressing ENTER.

This is the group to which schema access is being granted.

16 Enter the name of your import or export repository, or accept the default, `Siebel Repository`.

17 Enter the name of your repository file or accept the default, `mstrep.dat.`

The utility displays the values entered so far and prompts you as to whether you want to run the Upgrade Wizard.

- If the values are correct, launch the Wizard (`srvrupgwiz`) by entering `Y.`
- To change the values you previously entered, rerun the Configuration Wizard by entering `dbsrvr_config.ksh`. This allows you to reconfigure with different values.

NOTE: If a program or system error occurs and you need to rerun the Wizard, you can do so, starting at the point at which the wizard failed, by navigating to `$`*`SIEBEL_ROOT`*`/bin` and entering: `srvrupgwiz /m master_imprep.ucf`

If the repository import was successful, it displays a message that the repository has been imported. Otherwise, see the log files.

Reviewing the Log Files for Repository Import

The repository import process creates a number of log files within the subdirectory `siebsrvr\log` (`$SIEBEL_ROOT/log` under UNIX) that you must review for any errors. The log file names for subsequent processes include a numeric suffix that increments for additional logs files; for example, UpgWiz.log, UpgWiz_01.log, UpgWiz_02.log (under Windows) or srvrupgwiz.log, srvrupgwiz1.log, srvrupgwiz2.log (under UNIX). Further log files are created when the Upgrade Wizard encounters a problem and the user attempts a retry.

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

- imprep_prim.log
- imprep_lang.log

For more details on logs, refer to the discussion under "Reviewing the Log Files for Windows" on page 180 and "Reviewing the Log Files for UNIX" on page 184.

Acceptable Errors

The log files might include errors that are expected and benign. Compare any error messages found in the log files to the sample error messages in the errors.rtf file that is located in the db2390 subdirectory of the `dbsrvr` installation directory. (If a log file is not listed in the errors.rtf file, then there are no acceptable error messages for that log file.) No further action is required if the log files contain errors listed in the errors.rtf file.

NOTE: Only one of each type of error occurring in a particular log file appears in the errors.rtf file.

If you find errors not listed in the errors.rtf file, correct the condition that caused the errors, and rerun the Upgrade Wizard. If you cannot determine the cause of the errors on your own, contact Siebel Technical Support.

Do not review error numbers alone, because error numbers might change after installation of a new driver version. Instead, compare the actual error descriptions to find out which are acceptable errors for this platform.

CAUTION: Although other errors are rarely encountered, this review is critical. Certain errors, such as a failure to create indexes, might result in performance problems or anomalous behavior in Siebel eBusiness Applications.

Troubleshooting the Repository Import

One problem that can occur at this stage is the failure of the Siebel Repository import midway through the process. If this occurs, you must clean up the data from the failed import using Siebel Tools to delete the repository.

Postinstallation Tasks

Postinstallation tasks consist of the following activities:

- "Granting Table Privileges"
- "Loading Statistics" on page 200
- "Validating the Siebel Schema" on page 200
- "Installing Multilingual Seed Data" on page 202
- "Importing a New Language to Your Repository" on page 207
- "Populating the Siebel File System" on page 211

Granting Table Privileges

The database administrator must edit and execute the file grantstat.sql, located within the db2390 subdirectory of your Siebel Database Server installation directory (dbsrvr), to grant the Siebel group ID additional, required privileges on EIM tables.

To edit and run the grantstat.sql script

1 Locate the grantstat.sql script and edit it by replacing `&1` with the Siebel Schema Qualifier ID that you recorded in Appendix A, "Deployment Planning Worksheet."

2 Substitute `&2` with the security group ID or with any user ID.

3 Execute the grantstat.sql script using any method you are used to.

NOTE: If you are using multiple security groups, you must grant this privilege to each group.

Loading Statistics

Execution of the script loadstat.sql loads statistics required for operation of the Siebel application into the DB2 catalog.

To edit and run the loadstat.sql script

1. Locate the script loadstat.sql within the db2390 subdirectory.
2. Follow the instructions documented inside loadstat.sql to edit the file, making sure to change the variable *`SIEBELTABLECREATOR`* to the CREATOR for the Siebel Schema as defined in the DB2 catalog.
3. Execute the script using any method you are used to.

Validating the Siebel Schema

After installation and repository import, you should compare the physical database schema with the repository to make sure that there are no inconsistencies. This is accomplished using the Siebel Server utility dbchck.exe (Windows) or dbchck (UNIX), located within the binary subdirectory of your Siebel Server installation directory.

You can use the dbchck utility to validate data relationships, including foreign keys and the list of values. You can also use this utility when you have made changes to the extensibility of your Siebel database.

The most popular mode in which to run dbchck is using the option `/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 binary subdirectory of your Siebel Server installation directory, locate dbchck.exe or dbchck, depending on your OS platform.

2. If your midtier platform is Windows, delete the diccache.dat file before running dbchck. But, be sure that no EIM or Siebel Remote operations are running before deleting it.

   The dbchck utility creates a new diccache.dat file as a first step before proceeding with the integrity check. By deleting the dictionary cache file before starting, you make sure that dbchck validates against the Siebel Repository you specify.

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

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

   Environment variable scripts are located in the binary subdirectory of your Siebel Server installation directory.

4. Source the database profile.

5. Run dbchck using the following syntax:

   ```
   dbchck /u SADMIN /p password /t SIEBTO /r "Your Siebel Repository
   Name" /l dbchck.log /dict /all /s ODBC data source
   ```

   where:

   - `password` = login password to the database.
   - `Your Siebel Repository` = the repository that you want to compare against the physical data model; in the installation, the default value is `“Siebel Repository”`.

     NOTE: You must specify your repository name within quotation marks (“”) after the repository name parameter (`/r`).

- `ODBC data source` = the ODBC data source applicable to the repository.
- `(/t)` = the Siebel Schema Qualifier.

NOTE: To see a list of all 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 binary directory of your Siebel Server installation directory (siebsrvr).

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

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

Installing Multilingual Seed Data

If your organization deploys internationally and, therefore, requires data to be in multiple languages, you must install multilingual seed data (for example, lists of views, responsibilities, or system preferences). To do this, you add new language packs to your database after you have installed the base language for your database server. This populates the List of Values (LOV) with seed data in the new language.

Only after you successfully install seed data in your base language, can you add seed data in other languages to your database. Adding this seed data also adds new LOV data in the new language.

CAUTION: You cannot add secondary languages to the Siebel database for an Enterprise Server unless you have already installed them on the associated Siebel Server.

To install multilingual seed data under Windows

1. Follow the instructions under "Siebel Database Server Software Installation" on page 111, using the CD for the new language that you want to install.

2. In the Existing Installations Found screen, select the installation to which you want to add the language.

3. When installation is complete, launch the Database Server Configuration Utility.

4. When the Select Installation Operation screen appears (Step 7 on page 130), select Add a language to an existing Siebel Database, and then click Next.

   NOTE: To add seed data in a new language to your database, you must have already imported your repository in its base language.

   The Base Language screen appears and displays the languages you have installed for your Siebel Database.

5. In the Base Language screen, specify which of the languages you installed is the one in which you want to primarily run your database. (This is known as your base language.)

   To continue, click Next.

   NOTE: The Language Selection screen appears only if you have installed the files for multiple Siebel Language Packs. Installation of multiple Language Packs can occur either during your initial installation of Siebel 7 or at a later time.

6. In the Language Selection screen, specify the new language you are adding for this database.

   To continue, click Next.

   The ODBC Data Source Name screen appears.

7. Respond to the remaining prompts as you did when you originally completed the procedure, "Installing the Siebel Schema" on page 126, until the Repository Name screen appears.

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

To continue, click Next.

9 In the Configuration Parameter Review screen, review the configuration values you entered on the previous configuration screens:

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

A message box appears with the prompt:

```
To apply the configuration now, press “OK”.
To apply the configuration later, press “Cancel”.
The command line to apply the configuration later is
C:\sea704\siebsrvr\bin\siebupg.exe /m master_install_lang.ucf
```

NOTE: The last line in the prompt is generated dynamically based on the location of your Siebel Server installation.

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

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

The DB Server Configuration Utility displays the items to be executed or imported.

10. To begin, click OK.

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

    To verify that the installation was successful, review the log files. See “Reviewing the Log Files for Repository Import” on page 197.

TIP: Locate the file master_install_lang.ucf in the binary subdirectory of your Siebel Server installation directory and print it out, so that you have a record of the values you entered.

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

12. After you enable MLOV capability as described in *Global Deployment Guide*, proceed to “Importing a New Language to Your Repository” on page 207.

To install multilingual seed data under UNIX

1. Follow the UNIX instructions under “Siebel Database Server Software Installation” on page 111, using the CD for the new language that you want to install.

2. Follow the instructions for UNIX under “Performing a Standard Installation” on page 128 and when prompted for the installation operation (Step 7 on page 141), enter `2` to Add New Language to an Existing Database.

3. At the Language Selection prompt, specify which of the displayed languages you are adding on top of your base language for this database.

4. At the Base Language prompt, specify which language you want as your base language.

5. Respond to the remaining prompts as you did when you originally completed the procedure for “Installing the Siebel Schema” on page 126 whether you chose standard or custom installation.

6 When you are prompted for the repository name, accept the default (Siebel Repository) or enter another name.

7 Review the values for accuracy.

- If the values are correct, launch the Siebel Upgrade Wizard (srvrupgwiz) by entering `Y`.

  NOTE: If a program or system error occurs and you need to rerun the Siebel Upgrade Wizard, you can do so, starting at the point at which the wizard failed, by navigating to `$`*`SIEBEL_ROOT`*`/bin` and entering:

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

- To change the values you previously entered, rerun the Configuration Wizard by entering dbsrvr_config.ksh. This allows you to reconfigure with different values.

The DB Server Configuration Utility displays the items to be executed or imported.

8 To begin, click OK.

A message appears when the installation is complete.

To verify that the installation was successful, review the log files. See "Reviewing the Log Files for Repository Import" on page 197.

TIP: Locate the file master_install_lang.ucf in the binary subdirectory of your Siebel Server installation directory and print it out, so that you have a record of the values you entered.

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

10 After you enable MLOV capability as described in *Global Deployment Guide*, proceed to "Importing a New Language to Your Repository" on page 207.

Importing a New Language to Your Repository

After you successfully import your Siebel Repository in its base language, you can add additional languages to it.

By adding a new language to your repository, you populate rows of localizable information, which allows Siebel eBusiness Applications to better operate in the new language.

Regardless of how many Siebel eBusiness Applications you are using (such as Siebel Sales, Siebel Service, Siebel Marketing), you perform this step only once for each language you want to install.

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

To import a repository in a secondary language under Windows

1. After installing multilingual seed data ("Installing Multilingual Seed Data" on page 202) launch the DB Server Configuration Utility.

2. Follow the steps in "Importing and Exporting the Siebel Repository" on page 188 to Step 7 on page 189 (Select Repository Operation) and select:

   ```
   Add Language to an Existing Repository
   ```

   To continue, click Next.

3. In the Language Selection screen, specify the new repository language you are adding.

   To continue, click Next.

4. Follow the remaining procedures in "Importing and Exporting the Siebel Repository" on page 188 up to Step 12 on page 191.

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

   - **Import Repository Name.** Type the designated name for this Siebel Repository or accept the default.

- **Localized Repository Name.** If you are importing a secondary language repository on top of your base-language repository, this field appears in place of the Repository File Name field. Accept the default installation path and file name for this repository or type another valid installation path.

To continue, click Next.

6 In the Configuration Parameter Review screen, review the configuration values you entered on the previous configuration screens:

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

A message box appears with the prompt:

```
To apply the configuration now, press “OK”.
To apply the configuration later, press “Cancel”.
The command line to apply the configuration later is
C:\sea704\siebsrvr\bin\siebupg.exe /m master_imprep_lang.ucf
```

NOTE: The last line in the prompt is generated dynamically based on the location of your Siebel Server installation.

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

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

The DB Server Configuration Utility displays the items to be executed or imported.

7 To begin, click OK.

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

To verify that the installation was successful, review the log files. See "Reviewing the Log Files for Repository Import" on page 197.

TIP: Locate the file master_imprep_lang.ucf in the binary subdirectory of your Siebel Server installation directory and print it out, so that you have a record of the values you entered.

To import a repository in a secondary language under UNIX

1 Perform the steps described within the UNIX section of "Importing and Exporting the Siebel Repository" on page 188 up to the prompt for Repository Operations.

2 At the Repository Operations prompt, enter 2 for Add language to an existing Repository.

3 At the Choose Language prompt, enter the number corresponding to the secondary language you are adding.

NOTE: Siebel Systems does not support installation of any language other than U.S. English on top of a base Japanese, Simplified Chinese, Traditional Chinese, or Korean Language Pack.

4 Answer the remaining prompts with the values that you previously entered when you completed "Importing and Exporting the Siebel Repository" on page 188 for your base language until prompted for the Import Repository Name.

5. At the Import Repository Name prompt, accept the default name of the new repository or specify a new name.

6. Enter the name of your repository file name or accept the default, mstrep.dat.

   The utility displays the values entered so far and prompts you as to whether you want to run the Upgrade Wizard.

   - If the values are correct, launch the Wizard (srvrupgwiz) by entering `Y`.
   - To change the values you previously entered, rerun the Configuration Wizard by entering `dbsrvr_config.ksh`. This allows you to reconfigure with different values.

   NOTE: If a program or system error occurs and you need to rerun the Wizard, you can do so, starting at the point at which the wizard failed, by navigating to `$`*`SIEBEL_ROOT`*`/bin` and entering: `srvrupgwiz /m master_imprep_lang.ucf`

   To verify that the installation was successful, review the log files. See "Reviewing the Log Files for Repository Import" on page 197.

TIP: Locate the file master_imprep_lang.ucf in the binary subdirectory of your Siebel Server installation directory and print it out, so that you have a record of the values you entered.

Populating the Siebel File System

Specific files needed to run the Siebel File System, such as correspondence templates and Siebel Marketing files, are provided with the Siebel Database Server software. A subdirectory called files is created automatically when you install the Siebel Database Server.

You must populate the appropriate subdirectory of the Siebel File System with these file attachments after installing the Siebel Database Server and before running the Siebel Web Client.

To populate the Siebel File System directory

1. Copy the appropriate files from the files subdirectory of the Siebel Database Server software to the att subdirectory of the Siebel File System.
2. Copy the KB.kb file from the files subdirectory of the Siebel Database Server software to the cms directory within the Siebel File System.
3. Verify that the files are where they need to be.

Customizing a Development Environment 8

This chapter describes procedures for customizing development environments and then migrating those customizations to user-acceptance or production environments on DB2 UDB for OS/390 and z/OS.

Customization of Siebel eBusiness Applications within your development environment includes:

- Converting nonpartitioned tables to partitioned tables
- Changing views, business objects, applets, and tables
- Adding new columns to existing tables and adding new tables
- Modifying workflow policies and workflow processes

These tasks are documented in *Siebel Tools Reference* and *Siebel Workflow Administration Guide*. However, several procedures and concerns are specific to DB2 UDB for OS/390 and z/OS, so they are therefore included in this chapter.

This chapter assumes that you perform your development on either of the following:

- DB2 UDB on a Windows 2000, AIX, or Solaris machine
- DB2 UDB for OS/390 and z/OS on a partition of your zSeries machine reserved for development

In many cases, changes to user-acceptance and production (server) databases are made by a database administrator working within a change-management system. Therefore, this guide includes procedures for generating Data Definition Language (DDL) files that are later applied to databases.

Siebel Tools Considerations

There are certain considerations and procedures to follow in Siebel Tools when developing applications that run against databases on DB2 UDB for OS/390 and z/OS.

Setting Siebel Tools Configuration File Parameters

The following parameters in the [ServerDataSrc] section of the Siebel Tools application configuration file (tools.cfg) must be set for DB2 UDB for OS/390 and z/OS as shown in Table 16.

Table 16. Siebel Tools Configuration File Parameters

Parameter	Value
`TableOwner`	Schema qualifier
`MaxCursorSize`	-1
`PrefetchSize`	-1

Setting Database Options

When developing applications on any platform (for example, DB2 UDB) that will eventually be deployed on DB2 UDB for OS/390 and z/OS, you must set the proper database options in Siebel Tools. This enables all Siebel Tools features for DB2 UDB for OS/390 and z/OS and validates Siebel objects for OS/390 and z/OS requirements:

- The names of schema objects are limited to 18 characters.
- CHAR columns with length > 1 are allowed.
- Users can eliminate unused indexes.
- Users can reduce the size of VARCHAR fields.

To set database options for DB2 UDB for OS/390 and 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.

Specifying Table Grouping File Names

The complete paths for table grouping file names (storage control files) must be specified in the Apply dialog box when applying schema changes to a database.

Inactivating Unused Indexes

The standard Siebel data model has a large number of indexes that can degrade performance on DB2 UDB for OS/390 and z/OS. This is because DB2, unlike Oracle and Microsoft SQL Server, stores `NULL` as one of the indexed values. Each row with a `NULL` value for an indexed column gets an entry of its record identifier (RID) in the DB2 index. Because many Siebel tables are related to a large number of other tables, there are many indexes on foreign keys.

Where those related tables are not used, DB2 has the indexes for those foreign keys with one entry: `NULL`. This key entry has a long RID chain comprised of all the RIDS for every row in the table. There is maintenance overhead each time a row is inserted, and even more overhead when it is deleted. Therefore, users of DB2 UDB for OS/390 and 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.

Reducing Field Lengths of Unique Indexes to Avoid Truncation

Indexes on DB2 UDB for OS/390 and z/OS are limited to 255 characters. Indexes supplied by Siebel eBusiness Applications adhere to this limitation.

However, if you customize your development environment without setting the database options for DB2 UDB for OS/390 and 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 214.

When analyzing your business needs, if the unique indexes seem too long, Siebel Systems recommends that you reduce the field lengths.

Reducing VARCHAR Field Lengths

DB2 UDB for OS/390 and z/OS pads VARCHAR fields to their maximum lengths in indexes. For greatest flexibility, Siebel applications make extensive use of VARCHAR columns.

This might cause performance degradation for OS/390 and 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 must reduce the length of participating columns. To make better use of index space, users of DB2 UDB for OS/390 and z/OS are allowed to reduce VARCHAR lengths—where there is no application impact—using Siebel Tools.

You should analyze your business needs at installation or upgrade to determine whether it is necessary to reduce VARCHAR field lengths. Some examples are the following:

- FST_NAME and LAST_NAME, VARCHAR(50) columns in S_CONTACT. These belong to many indexes.
- ADDR, a VARCHAR(200) column in S_ADDR_PER.
- NAME, a VARCHAR(100) column in S_ORG_EXT.

Be aware that you also must make the same length reductions to columns denormalized from those you have reduced. One example is S_PER_RESP.PER_FST_NAME, which is denormalized from S_CONTACT.FST_NAME.

A denormalized column duplicates the data in a column in another (base) table for performance reasons. The table and column names of the duplicated column are specified in the Denormalization Path property of the Column object definition of the denormalized column. For more information on columns, see *Siebel Tools Reference*.

Using Character Large Objects (CLOBs)

Siebel objects defined as LONGVARCHARs can also be stored as character large objects (CLOBs) on DB2 UDB for OS/390 and z/OS. Siebel eBusiness Applications provide CLOBs that allow up to 32 KB of data for a single data element in a table row.

The default setting for these objects in the Siebel Schema is LONGVARCHAR, because of the performance and storage characteristics of CLOBs in a DB2 UDB for OS/390 and z/OS environment. However, as an exception to this rule, customers running Siebel Marketing should change the LONG VARCHAR column in the S_NOTE table to a CLOB.

The Siebel installation, upgrade, and migration processes create all auxiliary objects necessary to support CLOBs, so that their use is transparent to the user. For more information on the use of CLOBs, see *Upgrade Guide for Siebel eBusiness Applications on DB2 UDB for OS/390 and z/OS* for your operating system.

NOTE: CLOBs are turned off by default. They must be turned on for each table by setting `Clobs = Yes` in the storage control file.

If CLOBs are chosen for a table, the Siebel installation, upgrade, and migration processes create all auxiliary objects necessary to support CLOBs, so their use is transparent to the user.

NOTE: If you write your own scripts, it is recommended that you convert columns that contain scripts from LONGVARCHAR to CLOB to safeguard against space limitations that are inherent to the structure of OS/390 and z/OS.

For more information on the use of LONGVARCHARs and CLOBS, refer to the *Upgrade Guide* for your operating system.

Converting Nonpartitioned Tables to Partitioned Tables

Siebel eBusiness Applications support two scenarios for converting nonpartitioned tables to partitioned tables.

NOTE: In the procedures below, source table refers to your current nonpartitioned table, and target table refers to the destination partitioned table.

- **Source table and target table are defined in different table spaces.** If the source table and target table are defined *in different table spaces*, and the target table space is a new table space that does not yet exist in your current database, no special action is required. Run the two utilities available in the Siebel Software Configuration Utility (Migrate Development to Production, Synchronize Database Schema Definition) to automatically rebuild the table in the new partitioned table space and drop the original table.
- **Source table and target table are defined in the same table space.** If the source table and target table are defined *in the same table space* (for example, if you used the dbconfig.xls spreadsheet to convert an existing table space from nonpartitioned to partitioned), then you must perform one of the following procedures (as appropriate for your configuration) to convert a nonpartitioned table to a partitioned table on the same table space.

  NOTE: You must perform one of the following procedures, because the database operations do not support rebuilding a table space from nonpartitioned to partitioned. (This results in a validation failure and a halted operation.)

Perform the following steps if your database layout has a multiple-tables-per-table space structure, and you want to convert a nonpartitioned table to a partitioned table defined in the same table space.

To convert to a partitioned table for a multiple-tables-per-table space configuration

1 Review the Siebel objects naming convention that is used in the one-table-per-table space partitioned storage control file (storage_p_1to1.ctl) to determine the name of the table space that you want to move your nonpartitioned table to.

NOTE: Verify that the new table space name is unused.

2 Modify the storage file with the new table space name by using the dbconfig process (define it as a partitioned table space and define the table in this new table space).

3 Run the Siebel Software Configuration Utility: Migrate Development to Production. This process automatically rebuilds the source tables in the new partitioned table space and drops the original table.

Perform the following steps if your database layout has a single-table-per-table space structure, and you want to convert a nonpartitioned table to a partitioned table defined in the same table space.

To convert to a partitioned table for a one-table-per-table space configuration

1 Use your preferred database tool to manually move every nonpartitioned table (every table that you intend to convert) to a temporary table space, and then drop their original table spaces.

2 Use the dbconfig.xls utility to modify the storage control file, changing the mode of the table spaces to partitioned.

3 Run the Siebel Software Configuration Utility: Migrate Development to Production. This process automatically rebuilds the source tables in the new partitioned table space and drops the original nonpartitioned table.

4 Manually drop the temporary table space that you created in Step 1.

To convert nonpartitioned tables to partitioned tables using alternate tools

- Your DBA can use alternate tools (for example, BMC Change Manager or IBM Compare Utility) to convert nonpartitioned tables to partitioned table spaces. In this case, after changes have been applied, the last step is to change the schema version by executing the following SQL:

  ```
  update S_APP_VER
  
  set CUSTOM_SCHEMA_VER = char(integer(CUSTOM_SCHEMA_VER) + 1)
  ```

  This SQL is stored in bumpver.sql and can be executed outside of the Siebel application.

Creating Custom Extensions to the Siebel Schema

There are several different scenarios for Siebel schema extensions:

- Creating a small extension column, so the table fits into its existing table space
- Creating a large extension column, so the table has to be redefined to a larger page size and therefore a larger table space
- Creating an extension table

Creating a small extension column works as on other platforms and is documented in *Siebel Tools Reference*.

Creating a large extension column or an extension table requires the specification of a 16-KB or 32-KB table space:

- If developing on DB2 UDB for Windows or UNIX, you must enter a 16-KB or 32-KB table space in the Apply Schema screen. (See Step 4 on page 226.)
- If developing on DB2 UDB for OS/390 and z/OS, your DBA must edit the storage control file to specify in which database and table spaces the new extension column or table should reside.

For information on estimating storage needs, see Chapter 2, “Preparing for Implementation on the DB2 Host.”

About Schema Extension Tasks

The following tasks must be performed to extend the Siebel schema:

- Check out and lock the project to which the table being extended belongs. In most cases this is Newtable.

  If you want to import data into the new column using Enterprise Integration Manager (EIM), you must check out the project to which the appropriate interface table belongs. For more information, see the sections on EIM in *Siebel Tools Reference*.

- Update the logical schema definition in the development environment using Siebel Tools. This works as on other platforms and is documented in *Siebel Tools Reference* and *Developing and Deploying Siebel eBusiness Applications*.

  NOTE: Make sure that you have selected the check box on the Database tab of the Development Tools Options dialog under the View menu. This validates the objects' compliance with DB2 UDB for OS/390 and z/OS naming and sizing conventions.

- Apply the physical schema extensions to the development database:

  - If developing on DB2 UDB for Windows or UNIX, follow the procedures in"Applying Schema Extensions to the Development Database" on page 225.
  - If developing on DB2 UDB for OS/390 and z/OS, follow the procedures in "Migrating Customizations from Development to the Target Database" on page 228.

  In some companies, direct database extension is not allowed. Siebel Tools does not allow the Apply button to be disabled. The Apply process will fail if developers do not have proper database privileges (such as CREATEDBA and DBADM). For information about the Apply process, see "Applying Schema Changes to the Target Database" on page 234.

  If you do not want developers to extend databases, 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 228 to apply schema extensions to their development databases.

- Update and test configuration changes that apply to the extensions.

  See *Siebel Tools Reference*, *Developing and Deploying Siebel eBusiness Applications*, *Applications Administration Guide*, and other documentation on the *Siebel Bookshelf* CD-ROM.

Creating Storage Control File Objects for New Schema Objects

If you define a new table in Siebel Tools (for example, if you add a new extension table), you must add the table object definition to your storage control file in order for the table to be placed in the appropriate table space.

For example, if you create a new extension table called X_EXTENT, you must create a corresponding object definition in the storage control file:

```
[Object 6001]
Type=Table
Name=X_EXTENT
Database=SIEBDB20
Tablespace=S0600104
CLOBS=NO
```

NOTE: If you have different storage control files for your development environment and production environment, then you must add the new table object definition to both storage control files.

Applying Schema Extensions to the Development Database

Once your customizations are completed, you are ready to update your local DB2 UDB for Windows or UNIX environment.

NOTE: Siebel 7 does not support customized database triggers. If you have created customized triggers on your Siebel base tables, you must disable them before updating the logical database schema. You must re-create the triggers after the update is finished.

To update your local environment

1. In Siebel Tools, select the table from which you want to apply changes to the database.
2. Click Apply in the Object List Editor.

   A message box appears, alerting you that you are about to connect to a local database and asking if you want to continue.
3. Click OK.

   The Apply Schema message box appears.

4 Fill in the fields as shown in the following table, and then click Apply.

Field	Description
Tables	Select one of the following options from the drop-down menu: ■ **All.** Update the database to reflect all changes made to the dictionary. This option forces each database object to be compared with the data dictionary, and updated if required. ■ **Current Query.** Update the database to reflect modifications made to the tables in the current query only. ■ **Current Row.** Update 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.
Table groupings file	Specify the complete path for the storage control file. This file is provided by the DBA and is specific to your database.
Privileged user id	Enter the ID of a database user with CREATEDBA or DBADM privileges. The schema qualifier is read from `tools.cfg`.
Privileged 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.

Applying Schema Extensions to the Target Database

Once you have tested your schema customizations in the local environment, you can migrate them to the target user-acceptance or production database. You must perform the following tasks:

- "Preparing the Target Database"
- "Migrating Customizations from Development to the Target Database" on page 228
- "Applying Schema Changes to the Target Database" on page 234

Preparing the Target Database

Your DBA must complete the following actions before migrating the changes to the target database:

- Prepare the storage control file (for example, my_storage_file.ctl).

> **NOTE:** If you have created any new extension tables, your DBA must edit the storage control file to specify in which database and table spaces the new extension tables should reside.

- Ask all mobile users to synchronize.
- Make sure all connected clients are disconnected from the Siebel Database Server.
- Once all mobile user transactions have been merged and routed, stop the Siebel Enterprise Server.
- Perform a full backup of the database.

> **NOTE:** If you are changing the data type or length of custom extension columns that already have data in the target database, you must export that data before making the schema changes. After making the changes, import the data back into the target database.

Migrating Customizations from Development to the Target Database

CAUTION: Before you migrate your customizations from your development database to your target database, you must validate your storage control file. See "Validating the Storage Control File" on page 161.

When migrating customizations from your development source to your target database on the DB2 host, you can use one of two modes of execution:

- **Run DDL Automatically.** Select this to apply the DDL and DML required to create the Siebel Schema directly against the database, using an ODBC connection.
- **Generate DDL Into Files.** Select this to generate the DDL required to create the Siebel Schema.

  You must later use FTP or a similar file transfer program to transport the DDL (schema.sql and bumpver.sql) to the zSeries machine, where the DBA executes it using customary methods, for example SPUFI.

This section describes the DDL-generation mode of execution that is generally used under a change-management system.

NOTE: If your development environment is on DB2 UDB for OS/390 and z/OS, your DBA can also clone it to the target database. For more information, see "Cloning an OS/390 or z/OS Database" on page 236.

NOTE: Siebel 7 does not support customized database triggers. If you have created customized triggers on your Siebel base tables, you must disable them before migrating the database schema. You must re-create the triggers after the migration is finished.

To migrate the schema under Windows

1. Launch the Database Server Configuration Utility:

   Choose Start > Programs > Siebel Enterprise Server 7.x.x > Configure DB Server.

   or

   - **a** Open a command prompt.
   - **b** Go to the *SIEBSRVR_ROOT*\BIN directory.
   - **c** Enter the following command:

     ```
     ssincfgw -l ENU -v Y
     ```

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

2. In the Gateway Server Address screen, type the following values as you recorded them in your copy of Appendix A, “Deployment Planning Worksheet.”

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

   To continue, click Next.

3. In the Siebel Server Directory screen, perform either of the following tasks:

   - Accept the default value displayed in the Siebel Server Directory field. (This is the *SIEBSRVR_ROOT* directory, for example, D:\sea7xx\siebsrvr.)
   - Use the browse button to select an alternate directory path.

   To continue, click Next.

4. In the Siebel Database Server Directory screen, perform either of the following tasks:

   - Accept the default path displayed in the Siebel Database Server Directory field. (This is the *DBSRVR_ROOT* directory, for example, D:\sea7xx\dbsrvr.)

- Use the Browse button to select an alternate directory.

To continue, click Next.

5. In the RDBMS Platform screen, select the database platform you use, for example, IBM DB2 UDB for OS/390 and z/OS.

   To continue, click Next.

6. In the Siebel Database Operation screen, select Migrate Repository, and then click Next.

7. In the OBDC Data Source Name screen, type your source ODBC connection name, and then click Next.

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

9. In the Schema Qualifier screen, enter the schema qualifier in uppercase and security group ID from your copy of Appendix A, "Deployment Planning Worksheet," and then click Next.

10. In the Source Database Repository Name screen, type the source and target database repository names—typically Siebel Repository for both, and then click Next.

11. In the Target RDBMS Platform screen, select IBM DB2 UDB for OS/390 and z/OS, and then click Next.

12. In the Target Database ODBC Datasource screen, type your target ODBC connection name, and then click Next.

13. In the Target Database User Name screen, enter your target database username and password, confirm the password by typing it again, and then click Next.

14. In the Target Schema Qualifier screen, type the target schema qualifier in uppercase and the target group authorization ID, then click Next.

15. In the Migrate Repository Mechanism screen, select the Generate DDL Into Files, and then click Next.

16 In the DDL Output Directory screen, type the DDL output directory and storage control file, and then click Next.

For example, use `DBSRVR_ROOT\db2390\dboutput\dev2prod` and `DBSRVR_ROOT\db2390\my_storage_file.ctl`, respectively.

The Configuration Parameter Review screen appears.

17 Review the parameters, and then click Finish.

The runnow screen appears, giving you the choice of applying the configuration now or applying it later (and specifying the command line to apply it later).

18 Click OK to apply the configuration now.

The Siebel Upgrade Wizard launches.

19 Click OK to perform the migration.

The Siebel Upgrade Wizard runs, creating the following files:

- `schema.sql`
- `bumpver.sql`

To migrate the schema under UNIX

1 Source environment variables from `$SIEBEL_ROOT`.

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

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

NOTE: *`LANGUAGE`* is not the same as *`LANG`*; the former is the language in which the Configuration Wizard runs, while the latter is the code for the Language Pack you installed.

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

csh

```
setenv SIEBEL_ROOT New Value
```

```
setenv LANGUAGE New Value
```

ksh

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

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

2. Go to `$SIEBEL_ROOT /bin` and enter:

   ```
   dbsrvr_config.ksh
   ```

   This launches the Database Server Configuration Wizard.

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

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

4. Specify the path of your Siebel Server root directory, or accept the default by pressing ENTER.

5. Specify the path of your database server root directory, or accept the default by pressing ENTER.

6. Enter the number that corresponds to your database platform.

7. From the Select Database Operation menu, choose Migrate Repository (4).

8. Enter the number that corresponds to the language you are using, typically `1` for English.

9. Enter your source ODBC connection name.

10. Enter your source database username and password.

11. Enter the schema qualifier in uppercase from your copy of Appendix A, “Deployment Planning Worksheet.”

12. Specify the source database repository name, or accept the default by pressing ENTER.

13 Specify the target database repository name, or accept the default by pressing ENTER.

14 Select the target database platform by entering the number that corresponds to IBM DB2 UDB for OS/390 and z/OS.

15 Enter your target ODBC connection name.

16 Enter your target database username and password.

17 Enter the target schema qualifier in uppercase.

18 Enter the target group authorization ID, or accept the default ID, SSEROLE by simply pressing ENTER.

19 In the Select Installation Mechanism screen, select Generate DDL Into Files (`1`).

20 Specify the DDL output directory, for example `DBSRVR_ROOT/db2390/dboutput/dev2prod`.

21 Specify the storage control file, for example `DBSRVR_ROOT/db2390/my_storage_file.ctl.`

The configuration parameters are listed and you are asked if you want to run `srvrupgwiz`.

22 Enter Y for yes.

The Siebel Upgrade Wizard launches.

23 When the following message appears, enter Y to perform the migration.

```
Files schema.sql and bumpver.sql with the Siebel Schema
modifications have been generated in the DDL Output Directory.
Please select Yes to exit now and apply the files.
```

The following files are created:

- schema.sql
- bumpver.sql

Applying Schema Changes to the Target Database

After generating DDL into files, you must later use FTP or a similar file transfer program to transport the DDL (schema.sql and bumpver.sql) to the zSeries machine, where the DBA executes it using customary methods, for example, SPUFI.

Verify the schema changes.

Synchronizing Schema Definitions with the Existing Siebel Schema

The Synchronize Schema Definitions process is used to synchronize the repository definitions with the existing Siebel Schema database. Detailed information about this utility is provided in *Siebel Tools Reference*.

On DB2 UDB for OS/390 and z/OS this process depends on the existing database physical layout and the contents of the storage control file. The resulting database storage layout will fully reflect storage control file settings.

When you synchronize repository definitions with the existing Siebel Schema, the following data sources are accessed:

- Siebel Repository
- Storage control file
- DB2 catalog

To synchronize schema definitions

1. Launch the Database Server Configuration Utility:

   Choose Start > Siebel Enterprise Server 7.x.x > Configure DB Server.

   or

   a. Open a command prompt.

   b. Go to the `*SIEBSRVR_ROOT*\BIN` directory.

   c. Run the following command:

      ```
      ssincfgw -l ENU -v Y
      ```

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

2 Select the following options: Run Database utilities > Synchronize Schema Definitions > Generate DDL into files.

   When you generate DDL into files, two output files are generated in the default directory (`$`*`SIEBEL_ROOT`*`\dbsrvr\DB2390\DBOUTPUT\DDLSYNC`):

   - schema.sql
   - bumpver.sql

3 Ask your DBA to apply the output files.

Cloning an OS/390 or z/OS Database

Your DBA can clone your existing Siebel database to the target database. The DB2 system cloning process involves the following tasks:

- Extracting a storage control file from the existing Siebel schema.
- Using the newly extracted storage control file to generate a DDL file that your DBA will apply to create a clone of the Siebel schema.

To do this, you must run the database server configuration utility twice.

Extracting a Storage Control File from the Development Database

The first run of the database server configuration utility "reverse engineers" the existing Siebel schema by creating a storage control file (`.ctl`) file from the database.

For instructions on how to use this utility, see "Extracting a Storage Control File from the DB2 Catalog" on page 155.

Generating a DDL File from a Storage Control File

The second run of the database server configuration utility uses the newly created storage control file to generate a DDL file. This DDL file will be applied to the database by your DBA to create the Siebel schema clone.

To generate the DDL file under Windows

1 Launch the Database Server Configuration Utility in either of two ways:

Choose Start > Siebel Enterprise Server 7.x.x > Configure DB Server.

or

a Open a command prompt.

b Go to the *`SIEBSRVR_ROOT`*`\BIN` directory.

c Run the following command:

```
ssincfgw -l ENU -v Y
```

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

2 In the Gateway Server Address screen, type the following values as you recorded them in your copy of Appendix A, “Deployment Planning Worksheet.”

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

To continue, click Next.

3 In the Siebel Server Directory screen, perform either of the following tasks:

- Accept the default value displayed in the Siebel Server Directory field. (This is the *`SIEBSRVR_ROOT`* directory, for example, `D:\sea7xx\siebsrvr`.)
- Use the browse button to select an alternate directory path.

To continue, click Next.

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

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

To continue, click Next.

5 In the RDBMS Platform screen, select IBM DB2 UDB for OS/390 and zSeries.

To continue, click Next.

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

7 In the Select Installation Operation screen, select Install Siebel Database, and then click Next.

8 In the OBDC Data Source Name screen, type your source ODBC connection name, and then click Next.

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

10 In the Schema Qualifier screen, type the schema qualifier in uppercase and security group ID for your new database, and then click Next.

11 In the Select Installation Type screen, select the Customized Install, and then click Next.

12 In the Select Installation Mechanism screen, select the Generate DDL Into Files, and then click Next.

13 In the DDL Output Directory screen, type the following, and then click Next.

- DDL output directory, for example: `DBSRVR_ROOT\DB2390\dboutput\install`
- new storage control file, for example: `DBSRVR_ROOT\DB2390\my_storage_file.ctl`

The Configuration Parameter Review screen appears.

14 Review the parameters, and then click Finish.

The runnow screen appears, giving you the choice of applying the configuration now or applying it later (and specifying the command line to apply it later).

15 Click OK to apply the configuration now.

The Siebel Upgrade Wizard launches.

16 Click OK to perform the operation when the following message appears.

```
Files schema.sql and bumpver.sql with the Siebel Schema
modifications have been generated in the DDL Output Directory.
Please select Yes to exit now and apply the files
```

The following files are created:

- schema.sql
- bumpver.sql

NOTE: If you want to clone the Siebel schema into more than one database, copy the generated DDL file, replace the schema qualifier in the DDL file, and then save the file under a new name.

To generate the DDL file under UNIX

1 Source environment variables from $*SIEBEL_ROOT*.

SIEBEL_ROOT should be the path of your Siebel installation directory.

LANGUAGE should be set to the language in which the Configuration Wizard prompts appear, for example, enu for U.S. English.

NOTE: *LANGUAGE* is not the same as *LANG*; the former is the language in which the Configuration Wizard runs, while the latter is the code for the Language Pack you installed.

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

csh

```
setenv SIEBEL_ROOT New Value
```

```
setenv LANGUAGE New Value
```

ksh

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

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

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

```
dbsrvr_config.ksh
```

This launches the Database Server Configuration Wizard.

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

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

4 Specify the path of your Siebel Server root directory, or accept the default by pressing ENTER.

5. Specify the path of your database server root directory, or accept the default by pressing ENTER.
6. Enter the number that corresponds to your database platform.
7. In the Select Database Operation menu, choose Install Database (`1`).
8. In the Installation Operation screen, select Install Siebel Database (`1`).
9. Enter the number that corresponds to the language you are using, typically `1` for English.
10. Enter your source ODBC connection name.
11. Enter your database username and password, and then confirm the password by typing it again.
12. Enter the schema qualifier for your new database in uppercase.
13. Enter the group ID for your new database, or accept the default (SSEROLE) by pressing ENTER.
14. In the Installation Type screen, select Customized Install (`2`).
15. In the Installation Mechanism screen, select Generate DDL Into Files (`1`).
16. Specify the DDL output directory, for example *`DBSRVR_ROOT`*`/DB2390/dboutput/install`.
17. Specify the new storage control file, for example *`DBSRVR_ROOT`*`/DB2390/my_storage_file.ctl`.

    The configuration parameters are listed and you are asked if you want to run `srvrupgwiz`.

18 Enter Y for yes.

The Siebel Upgrade Wizard launches.

19 Enter Y to perform the operation when the following message appears.

```
Files schema.sql and bumpver.sql with the Siebel Schema
modifications have been generated in the DDL Output Directory.
Please select Yes to exit now and apply the files
```

The following files are created:

- schema.sql
- bumpver.sql

NOTE: If you want to clone the Siebel schema into more than one database, copy the generated DDL file, replace the schema qualifier in the DDL file, and then save the file under a new name.

Migrating Rules and Other Data

After you have successfully migrated your source schema to your target database, you must copy data from your development environment. Such data might include:

- Modified files, such as the Siebel repository file (.srf), Web templates, image files, and cascading style sheets
- Transactional data, such as accounts, contacts, and opportunities
- Setup data, such as employees, positions, and responsibilities
- Program data, specifically Assignment Manager rules and Workflow processes and policies, and personalization rules and expressions

For information on copying such data, see Chapter 10, "Migrating Data Using Siebel EIM," and *Developing and Deploying Siebel eBusiness Applications*.

Applying Schema Changes to Other Local Databases

For information on upgrading local databases, see *Siebel Tools Reference* and *Developing and Deploying Siebel eBusiness Applications*.

Customizing Applications Using Assignment and Workflow Rules

There are two additional ways of customizing applications besides using Siebel Tools.

Siebel Assignment Manager allows you to create business rules that automatically assign entities such as opportunities, service requests, or activities to the most qualified individuals. For more information, see *Siebel Assignment Manager Administration Guide*.

Siebel Workflow provides a graphical interface for designing and implementing business processes and user interactions. Workflow processes define the steps to automate business processes, such as sending email. Workflow policies trigger processes when they detect certain conditions, such as an opportunity being assigned. For more information, see *Siebel Workflow Administration Guide*.

To activate rules—whether they are assignment or workflow—you must run the following Siebel Server components:

- **Generate Triggers (GenTrig).** Allows you to create database triggers.

  GenTrig writes database triggers to a SQL file, such as TRIGGER.SQL. On the DB2 host, the SQL file is executed manually by a DBA. For manual execution of TRIGGER.SQL, run GenTrig with the EXEC parameter set to FALSE.

  NOTE: For DB2, GenTrig does not log into the database as the table owner. Instead, it logs in as the privileged user. All triggers generated are qualified with the schema qualifier. When starting the GenTrig component, users are prompted for the schema qualifier.

- **Workflow Monitor.** Uses these database triggers to identify which records might match policy conditions.

For information on activating rules, see *Developing and Deploying Siebel eBusiness Applications*.

Maintenance Considerations for zSeries 9

This chapter describes postinstallation maintenance tasks that you can perform to improve performance of Siebel eBusiness Applications on DB2 UDB for OS/390 and z/OS.

Reorganizing Table Spaces, Partitions, and Indexes

It is recommended that you reorganize tables spaces, partitions, and indexes that have a tendency to become disorganized or 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, partitions, and indexes, because disorganization and fragmentation of these objects can affect performance significantly. (For a list of Siebel tables to monitor for reorganization, see Table 7 on page 62.)

You should not reorganize `S_ESCL_LOG`, `S_DOCK_INIT_ITEM`, and all `S_DOCK_INITM_%%` tables (where `%` is a digit).

You should not use 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) then loadstat.sql. For more information on how to run these utilities, see “Updating Statistics” on page 246.

CAUTION: You should not use online REORG of Siebel repository table spaces. To avoid adversely affecting the Siebel application, perform reorganization of Siebel repository tables paces during scheduled application downtime.

Updating Statistics

The DBA should update statistics on table spaces that contain EIM interface tables and base tables (including table spaces that contain extension tables and repository tables) when there has been a change of 20 percent or more in the row distribution. It is usually not necessary to update statistics on all of the table spaces, only on those containing tables that have changed.

Execution of the rstat390 utility from midtier executes RUNSTATS on the DB2 host. The rstat390 script is located within the binary subdirectory of the *SIEBSRVR_ROOT* directory. For rstat390 help, use option /h.

NOTE: As far as possible, you should update statistics only for table spaces that contain changed tables, not for all table spaces, to save time and prevent locking problems.

CAUTION: Never use RUNSTATS to update statistics for table spaces for S_DOCK_INIT_ITEM, S_DOCK_INITM%%, S_ESCL_LOG, S_CTLG_CAT, and S_OPTY. If you do so in error, run loadstat.sql afterwards.

You should update statistics only when there is little activity on the system, such as after midnight. If you run this utility while users are accessing and updating the Siebel Database, lock contention can occur. When this happens, an error message is generated, for example:

```
ODBC error S1000 in SQLExecDirect: [IBM][CLI Driver][DB2/6000]
SQL2310N The utility could not generate statistics. Error "-911"
was returned.
```

This does not harm your database, but the RUNSTATS job will have to be rerun for any table for which this type of error was generated, because statistics were not updated for that table.

You can execute RUNSTATS on an active system if you specify shrlevel change as an option. This allows concurrent access while the RUNSTATS utility executes.

You should execute RUNSTATS and then loadstat.sql following reorganization of any fragmented table spaces and indexes. For more information on this topic, see "Reorganizing Table Spaces, Partitions, and Indexes" on page 245.

You can also execute RUNSTATS directly from the DB2 host.

CAUTION: Because the rstat390 and RUNTSTATS utilities can overwrite statistics loaded by Siebel applications, you should always execute loadstat.sql after executing rstat390. Otherwise, valuable statistics might be lost.

About Cursor Close

Cursor close allows you to regulate system resource utilization by Siebel clients in an OS/390 or z/OS environment by setting cursor configuration parameters.

NOTE: If you want to configure cursor close, a white paper on that topic is available on Siebel SupportWeb.

The cursor configuration parameters specify the number of database rows a Siebel client cursor can set from a user-entered query or a Siebel operation that generates a SQL query. The number of rows is directly related to the length of time a Siebel Web Client occupies an active DB2 thread.

The cursor configuration parameters `DSMaxCursorSize` and `DSPreFetchSize` are read at startup from the following places:

- Enterprise profile configuration parameters on the Siebel Server for Siebel Web Clients
- Application configuration (`.cfg`) file for Siebel Dedicated Web Clients and Siebel Mobile Web Clients

The default value for both `DSMaxCursorSize` and `DSPreFetchSize` is `48`.

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 user connecting to DB2 UDB for OS/390 and z/OS. These guidelines are not mandated, but provide a balance between system resource utilization, user responsiveness, and productivity. Your enterprise might have unique requirements beyond the scope of this chapter. For assistance, contact Siebel Expert Services.

Migrating Data Using Siebel EIM 10

This chapter describes special considerations for customers using Siebel Enterprise Integration Manager (EIM) to import, export, update, merge, or delete data within Siebel eBusiness Applications on DB2 UDB for OS/390 and z/OS.

Before using EIM, familiarize yourself with *Siebel Enterprise Integration Manager Administration Guide*. This chapter only discusses those elements of EIM operation specific to running EIM on DB2.

Setting Up EIM for DB2

There are many ways to optimize EIM data throughput. EIM is similar in function to other DB2 applications, so your DBA should use regular performance monitoring tools to monitor and fine-tune performance.

You should review and tune performance for each set of business data you load with EIM, for example, Opportunities, Contacts, or Products.

Choosing an Appropriate Partitioning Scheme

To achieve the best system throughput, it is recommended that you partition base tables and EIM tables and set up parallel EIM processes to efficiently exploit partitioned EIM table spaces.

When a table is created on a partitioned table space, the table is assigned a partitioning key composed of one or more columns. EIM tables by default use a clustering U1 index in the order of IF_ROW_BATCH_NUM, ROW_ID. In some cases, you can obtain better performance by partitioning EIM tables based on the partitioning key of the target base table, for example, on IF_ROW_BATCH_NUM plus the partitioning key of the corresponding target table.

NOTE: The values in this key should not be updated. Changes to values in the partitioning key might require unloading and reloading or performing a REORG on the data to resolve performance issues.

For more information on DB2 partitioning, see Chapter 4, “Understanding Siebel Partitioning.”

Considering the Effect of EIM on Table Partitioning

The mechanism by which EIM generates the ROW_ID can result in an uneven distribution of data if you use EIM with one of the default partitioning schemes to import data into a base table. See “EIM Considerations for Table Partitioning” on page 102 for recommendations about how to avoid uneven distribution of data in a partitioned table that has data imported by EIM.

Planning Data Mapping

Data mapping—an important part of preparing to run EIM—is particularly important on DB2 due to the way DB2 stores data in tables and indexes. Your data mapping should take into account such factors as partitioning, look-ups during the load phase, and searches for online transactions.

Only Load Necessary Data

The Siebel eBusiness Applications Data Model provides for many possible business needs and configurations. Many of the tables, columns, and indexes provided with Siebel eBusiness Applications might not be relevant to your business. Because DB2 stores information in all columns, including NULL columns, you should carefully consider which data should be imported into your Siebel application as part of your initial planning. Unnecessary data increases the size of tables and indexes, which can affect performance.

For information on dropping columns and indexes and reducing column sizes using Siebel Tools, see Chapter 8, "Customizing a Development Environment."

Do Not Load MS_IDENT

The MS_IDENT column exists in every EIM table and is specific to DB2 UDB for OS/390 and 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. This can particularly be an issue when migrating data from a Siebel application on DB2 for UNIX or Windows to a Siebel application on DB2 UDB for OS/390 and z/OS.

Presorting Data During Initial Load

When loading the EIM tables with legacy data, sort this data on the partitioning key that will be used on the target base table. When sorting string data, use the same character-collating sequence used in the target database. During this load, you should also preassign and load batch numbers corresponding to the partitioning scheme of the target base tables when possible. This is particularly useful for those processes such as Accounts or Contacts, where the partitioning key is based on business data rather than on a generated key. This generally improves EIM throughput, as the data is then loaded sequentially and by partition into the target base table.

TIP: To further improve performance, once the EIM tables are loaded with partitioned data, you can unload this data in clustering order, then reload it.

Choosing an Optimal EIM Batch Size

Users can regulate EIM commit frequency in several ways, including:

- Setting `COMMIT EACH TABLE = TRUE`.
- Setting `COMMIT EACH PASS = TRUE`.
- Adjusting batch size.

Of these, the batch size most directly influences EIM performance.

The size of EIM batches can significantly affect throughput. This occurs as a result of the number of commits that EIM must execute. While each commit is CPU-intensive, commits release page locks, thereby releasing memory and avoiding lock escalation and time-outs.

While every installation is different, you should perform initial testing with large batch sizes (except DELETE EXACT, which normally runs better with smaller batch sizes). You might want to use this as a starting point for your own processes and modify batch size based on results.

NOTE: Processes that populate child tables might violate the maximum number of locks allowed by the system or cause contention for resources in those child tables. To avoid this, reduce the batch size of these processes.

Choosing an Optimal Number of Parallel Streams

A rule to determine how many batches can run simultaneously is that the combined number of parallel streams should keep the CPU 100% busy. These can either be multiple EIM processes running simultaneously or the same EIM process repeated against multiple partitions of the same table.

Running EIM processes in parallel on a DB2 database with the default setting of `UPDATE STATISTICS = TRUE` can cause a deadlock when multiple EIM processes access the same interface table simultaneously. To avoid a deadlock, set `UPDATE STATISTICS = FALSE` in the EIM configuration file.

A similar problem might occur if parallel processes access the same child tables while updating foreign keys. Therefore, analyze your EIM mappings and processes and do not run these processes in parallel. Alternatively, you can try altering the locking level from Page to Row.

Optimizing DB2 Database Logging

The number of parallel streams that the DB2 UDB for OS/390 and z/OS environment can support is often limited more by the DB2 logging throughput rate than anything else. The commit process waits for a successful write to the active log and, if there is contention on the logs, performance suffers. Possible solutions to ease bottlenecks caused by logging performance include:

- Increase the number of output buffers for writing active log datasets. The size of the output buffer can be changed in `DSNZPARM OUTBUFF`.
- Increase the active log size. If logs fill up faster than they can be archived, performance suffers.
- Determine the optimal number of DB2 active logs based on peak EIM loads plus a cushion.
- Place the active log data sets on separate dedicated volumes.
- Place archive logs on a virtual tape system (VTS), or disk if VTS is not available.
- Define all table spaces (base tables and interface tables) with compression, which reduces logging, and decreases I/O.
- Consider data sharing when a single subsystem cannot handle the logging tasks.

Using the RUNSTATS Utility

If EIM performance degrades over time, consider running RUNSTATS against the base tables to update the optimizer statistics.

If the parameter `UPDATE STATISTICS` is set to `TRUE` in your `.ifb` file, the EIM process invokes the DSNUTILS stored procedure. This allows EIM to perform RUNSTATS dynamically on temporary columns to get optimal statistics numbers in the DB2 catalog. The IBM DSNUTILS stored procedure must be installed and the OS/390 WLM manager should be enabled.

To execute EIM with `UPDATE STATISTICS` set to `TRUE,` the user account must have EXECUTE privileges on the DSNUTILS stored procedure and SYSADM, STATS, or DBADM privileges on the tables accessed.

NOTE: Set `UPDATE STATISTICS = TRUE` only once to collect proper statistics for each given EIM process. As soon as this is done, reset `UPDATE STATISTICS` to `FALSE` for that process.

For more information on RUNSTATS, see Chapter 9, “Maintenance Considerations for zSeries.”

Reviewing the SQLPROFILE Log

The `SQLPROFILE` parameter in the header of your EIM configuration (`.ifb`) file designates the file to which EIM writes an analysis of the worst-performing SQL queries overall (by total time) and the worst-performing statements in each step of the EIM process (also by total time), for example:

```
[Siebel Interface Manager]

      PROCESS = Import Products

      SQLPROFILE = topsql.log
```

Review this log file after the test run of each EIM process to determine where potential bottlenecks exist in your process. Based on the results of this log, either adjust mapping and IFB parameters, or perform normal DB2 tuning (such as adding an index to improve the access path).

Improving EIM Performance During the Import Process

The following procedures and techniques can improve EIM performance during import processes, particularly during your initial load of data into Siebel eBusiness Applications.

You should follow the recommended import order as described in *Siebel Enterprise Integration Manager Administration Guide* for both initial and ongoing EIM loads.

Using the Default Column to Improve Throughput

If certain values in your database will be constant across all rows of a table, use the `DEFAULT COLUMN` parameter to fill these rather than loading them through the EIM tables to improve throughput.

Tuning Your EIM Import Process

Review and tune your EIM process for each set of business data you load, for example, Opportunities, Contacts, or Products.

To tune each EIM import process

1. Load a single batch with the following flags set in your configuration file:

    - `Error flag = 1`
    - `SQL flag = 8`
    - `Trace flag = 3`

    This produces a log file with SQL statements and information about how long each statement took to execute.

2. Identify SQL statements that might be taking too long to execute.

3. Using the DB2 EXPLAIN utility, populate the explain table with information on how the DB2 Optimizer is executing each of these SQL statements. Pay particular attention to any changes in the default optimization level in the EIM log.

4. Based on the Access Plan and other information from the EXPLAIN output, determine required indexes for EIM and base tables.

5. Run EIM with parameters to reach each partition and repeat single process tuning.
6. Perform `.ifb` file tuning for each process. Remove unnecessary foreign keys.
7. Execute the REORG utility on base tables to build a compression dictionary.
8. Perform parallel process tuning.
9. Perform buffer pool tuning.
10. Execute STOSPACE on base tables. This collects storage information on DB2 objects.
11. Perform RUNSTATS on base and EIM tables.
12. Perform EXPLAINs to verify access path (required indexes).
13. Modify statistics if required.
14. Drop unused indexes on base and EIM tables.
15. Check space and resize table spaces if necessary.

Initial Load Considerations

There are certain considerations that are only applicable during an initial load of data into your Siebel application.

Drop Unused, Non-Unique Indexes

The initial load is typically a database-intensive process. Each row that is inserted into the base table requires modifications not only to the table itself, but to all its affected indexes. However, most of these indexes are never used during an initial load process.

Performance can usually be improved by determining which base table indexes are unused by EIM during the import process and dropping them prior to the initial load. These indexes must later be recreated by your DB2 database administrator (DBA).

Dropping Unused Triggers

Dropping unused triggers can improve import performance by reducing the system overhead used to verify trigger constraints. You can use the Generate Triggers (GenTrig) component from Siebel Server Tasks to drop the triggers and to re-create them later on. This component has to be in the "Enabled" state. For details on how to work with Generate Triggers, see *Siebel Workflow Administration Guide*.

NOTE: If you are employing partitioning, do not drop triggers that are used for partitioning purposes. If you do your EIM process fails, or it inserts all the data in a single partition.

Optimizing Free Space Parameters

If you will use your Siebel application primarily for queries, updates and deletions, you should alter your table spaces and indexes to provide optimal insert and update performance.

To improve the EIM import and update SQL performance, set PCTFREE for table spaces and indexes to a value of 20 or 30 prior to the EIM initial load, and maintain these settings at 20 or 30 for subsequent loads.

To improve the insert performance on base tables during EIM initial load

1. Set PCTFREE to zero for data and sequential index.
2. Set FREEPAGE to zero for data and index.
3. Use Page Lock if you are sequentially inserting many rows or pages.
4. For a data sharing environment, set the member cluster option to reduce spacemap and data page P-Lock contention.

You should alter base table spaces to PCTFREE and FREEPAGE values of 20 or 30 for subsequent import processes. You should also load or reorganize any altered items before the new values take effect.

The following example demonstrates how to set these values for your table spaces and indexes. Substitute your own table space and index names for your implementation.

```
ALTER TABLESPACE SIEBDB01.H0401000 PART 1 PCTFREE 20;
ALTER TABLESPACE SIEBDB01.H0401000 PART 2 PCTFREE 20;


ALTER INDEX SIEBDB01.S_ADDR_ORG_M6          PCTFREE 20;
ALTER INDEX SIEBDB01.S_ADDR_ORG_P1          PCTFREE 20;
ALTER INDEX SIEBDB01.S_ADDR_ORG_P99 PART 1 PCTFREE 20;
ALTER INDEX SIEBDB01.S_ADDR_ORG_P99 PART 2 PCTFREE 20;
```

Disabling Transaction Logging

Siebel transaction logging is unnecessary during an initial EIM load, and should be disabled by setting `LOG TRANSACTIONS = FALSE` in your `.ifb` file. For more information on transaction logging in EIM, see *Siebel Enterprise Integration Manager Administration Guide*.

Considerations for EIM Tables

Consider the following issues when working with EIM tables.

Updating the S_LST_OF_VAL Table

The BU_ID column in the S_LST_OF_VAL base table can have only one or very few distinct values. When this happens, the DB2 optimizer often needlessly performs a table scan through all rows in the S_LST_OF_VAL table.

To avoid this problem and speed up the query, modify the statistics data by running the following SQL statements:

```
update sysibm.sysindexes set firstkeycardf=1000 where

name='S_LST_OF_VAL_M2';

update sysibm.syscolumns set colcardf = 1000 where

tbname='S_LST_OF_VAL' and name='BU_ID';
```

NOTE: Depending on the data you are working with, you might need to run other SQL statements ahead of time.

Resolving Performance Degradation During the Load Process

When executing the EIM load process, performance might degrade noticeably over time, after subsequent batch loads. This performance degradation is often due to outdated statistics on the tables being loaded. To resolve the problem, update statistics on the target base tables.

Resolving Errors in the EIM Process

Perform the steps below to resolve errors that might occur during your EIM process:

- If EIM aborts with the following generic error, then the user account running EIM does not have EXECUTE privileges on the DSNUTILS stored procedure:

  ```
  DSNU060I database_alias DSNUGMAP - USER username NOT AUTHORIZED
  FOR RUNSTATS UTILITY ON
  ```

  See "Using the RUNSTATS Utility" on page 253 for information about required permissions. Alternatively, you can run EIM with the following setting:

  ```
  UPDATE STATISTICS = FALSE
  ```

- If your EIM process fails with the following error message, then the DSNZPARM parameter RETVLCFK might be set incorrectly.

  ```
  EIM-00205 Failed To Load the Application Dictionary
  ```

  Siebel eBusiness Applications require the setting `RETVLCFK=NO`. For information on DSNZPARM settings see Chapter 2, "Preparing for Implementation on the DB2 Host."

- If EIM terminates during a DELETE EXACT process involving deletion of child records, the following error indicates that the maximum allowable locks were exceeded:

  ```
  EIM Fails - Max Locks Exceeded on DELETE EXACT
  ```

  This causes the child records to be orphaned, because the delete to the parent table has already been committed.

  You can avoid this error by specifying the following in your `.ifb` file:

  ```
  COMMIT EACH PASS = FALSE

  COMMIT EACH TABLE = FALSE

  ROLLBACK ON ERROR = TRUE
  ```

Deployment Planning Worksheet A

The Deployment Planning Worksheet should be photocopied and a copy given to each member of the deployment team. Please do not write on the master copy in this Appendix. You must make copies of this master each time you install a new Siebel Enterprise Server.

Master Worksheet

Team Lead Summary

Deployment Team Lead: ______________________________

Siebel Administrator: ______________________________

Privileged User: ______________________________

DB2 Systems Programmer (SYSADM): ______________________________

DB2 Database Administrator (DBADM): ______________________________

Security Administrator: ______________________________

OS/390 and 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: ____________________

Enterprise Server Name: ____________________

Siebel Server Name: ____________________

Path to Siebel File System: ____________________

Database Alias: ____________________

Schema Qualifier ID (Max. 8 characters)	**Security Group ID**	
____________________	____________________	
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/CSSID**	
____________________	____________________	

Optimizing Performance with Buffer Pools B

To reduce synchronous I/O and increase performance levels, you should use the following buffer pool settings. These settings are based on tests performed in a closed system, so they might not increase performance in every environment. Adjust buffer pool size if memory usage increases or if you remove indexes.

NOTE: `ALTER` to the `BUFFERPOOL` and `START` and `STOP` the databases are DB2 commands, while `ALTER` to the objects is a SQL statement.

Zeroing the Buffer Pools

The buffer pools should be set to use data spaces. Failure to use data spaces can result in over use of storage by the DBM1 address space and cause the DB2 subsystem to fail. To change to data spaces, you must first zero the buffer pool allocations:

```
-ALTER BPOOL(BP2) VPSIZE(0)

-ALTER BPOOL(BP10) VPSIZE(0)

-ALTER BPOOL(BP12) VPSIZE(0)

-ALTER BPOOL(BP14) VPSIZE(0)

-ALTER BPOOL(BP16) VPSIZE(0)

-ALTER BPOOL(BP17) VPSIZE(0)

-ALTER BPOOL(BP18) VPSIZE(0)

-ALTER BPOOL(BP22) VPSIZE(0)

-ALTER BPOOL(BP23) VPSIZE(0)

-ALTER BPOOL(BP25) VPSIZE(0)

-ALTER BPOOL(BP26) VPSIZE(0)

-ALTER BPOOL(BP27) VPSIZE(0)
```

```
-ALTER BPOOL(BP30) VPSIZE(0)
-ALTER BPOOL(BP32) VPSIZE(0)
-ALTER BPOOL(BP33) VPSIZE(0)
-ALTER BPOOL(BP34) VPSIZE(0)
-ALTER BPOOL(BP35) VPSIZE(0)
-ALTER BPOOL(BP36) VPSIZE(0)
-ALTER BPOOL(BP37) VPSIZE(0)
-ALTER BPOOL(BP38) VPSIZE(0)
-ALTER BPOOL(BP39) VPSIZE(0)
-ALTER BPOOL(BP40) VPSIZE(0)
-ALTER BPOOL(BP41) VPSIZE(0)
-ALTER BPOOL(BP42) VPSIZE(0)
-ALTER BPOOL(BP43) VPSIZE(0)
-ALTER BPOOL(BP44) VPSIZE(0)
-ALTER BPOOL(BP45) VPSIZE(0)
-ALTER BPOOL(BP46) VPSIZE(0)
-ALTER BPOOL(BP47) VPSIZE(0)
-ALTER BPOOL(BP48) VPSIZE(0)
-ALTER BPOOL(BP49) VPSIZE(0)
-ALTER BPOOL(BP16K0) VPSIZE(0)
-ALTER BPOOL(BP16K1) VPSIZE(0)
-ALTER BPOOL(BP16K4) VPSIZE(0)
-ALTER BPOOL(BP16K8) VPSIZE(0)
-ALTER BPOOL(BP16K9) VPSIZE(0)
-ALTER BPOOL(BP32K1) VPSIZE(0)
-ALTER BPOOL(BP7) VPSIZE(0)
-ALTER BPOOL(BP32K) VPSIZE(0)
```

Changing the Buffer Pools to Data Spaces

After deallocating the buffer pools, modify the type:

```
-ALTER BPOOL(BP2) VPTYPE(D)

-ALTER BPOOL(BP10) VPTYPE(D)

-ALTER BPOOL(BP12) VPTYPE(D)

-ALTER BPOOL(BP14) VPTYPE(D)

-ALTER BPOOL(BP16) VPTYPE(D)

-ALTER BPOOL(BP17) VPTYPE(D)

-ALTER BPOOL(BP18) VPTYPE(D)

-ALTER BPOOL(BP22) VPTYPE(D)

-ALTER BPOOL(BP23) VPTYPE(D)

-ALTER BPOOL(BP25) VPTYPE(D)

-ALTER BPOOL(BP26) VPTYPE(D)

-ALTER BPOOL(BP27) VPTYPE(D)

-ALTER BPOOL(BP30) VPTYPE(D)

-ALTER BPOOL(BP32) VPTYPE(D)

-ALTER BPOOL(BP33) VPTYPE(D)

-ALTER BPOOL(BP34) VPTYPE(D)

-ALTER BPOOL(BP35) VPTYPE(D)

-ALTER BPOOL(BP36) VPTYPE(D)

-ALTER BPOOL(BP37) VPTYPE(D)

-ALTER BPOOL(BP38) VPTYPE(D)

-ALTER BPOOL(BP39) VPTYPE(D)

-ALTER BPOOL(BP40) VPTYPE(D)

-ALTER BPOOL(BP41) VPTYPE(D)

-ALTER BPOOL(BP42) VPTYPE(D)

-ALTER BPOOL(BP43) VPTYPE(D)

-ALTER BPOOL(BP44) VPTYPE(D)

-ALTER BPOOL(BP45) VPTYPE(D)
```

```
-ALTER BPOOL(BP46) VPTYPE(D)

-ALTER BPOOL(BP47) VPTYPE(D)

-ALTER BPOOL(BP48) VPTYPE(D)

-ALTER BPOOL(BP49) VPTYPE(D)

-ALTER BPOOL(BP16K0) VPTYPE(D)

-ALTER BPOOL(BP16K1) VPTYPE(D)

-ALTER BPOOL(BP16K4) VPTYPE(D)

-ALTER BPOOL(BP16K8) VPTYPE(D)

-ALTER BPOOL(BP16K9) VPTYPE(D)

-ALTER BPOOL(BP32K1) VPTYPE(D)

-ALTER BPOOL(BP7) VPTYPE(D)

-ALTER BPOOL(BP32K) VPTYPE(D)
```

Setting the Buffer Pool Sizes

These sizes are guidelines only and should be monitored to make sure that they work correctly in your subsystem. You can monitor buffer pool sizing, using the DISPLAY BUFFERPOOL command.

```
-ALTER BPOOL(BP0) VPSIZE(15000)

-ALTER BPOOL(BP2) VPSIZE(50000)

-ALTER BPOOL(BP10) VPSIZE(15000)

-ALTER BPOOL(BP12) VPSIZE(5000)

-ALTER BPOOL(BP14) VPSIZE(5000)

-ALTER BPOOL(BP16) VPSIZE(15000)

-ALTER BPOOL(BP17) VPSIZE(5000)

-ALTER BPOOL(BP18) VPSIZE(5000)

-ALTER BPOOL(BP22) VPSIZE(25000)

-ALTER BPOOL(BP23) VPSIZE(15000)

-ALTER BPOOL(BP25) VPSIZE(15000)

-ALTER BPOOL(BP26) VPSIZE(15000)
```

```
-ALTER BPOOL(BP27) VPSIZE(15000)
-ALTER BPOOL(BP30) VPSIZE(90000)
-ALTER BPOOL(BP32) VPSIZE(20000)
-ALTER BPOOL(BP33) VPSIZE(20000)
-ALTER BPOOL(BP34) VPSIZE(20000)
-ALTER BPOOL(BP35) VPSIZE(20000)
-ALTER BPOOL(BP36) VPSIZE(10000)
-ALTER BPOOL(BP37) VPSIZE(20000)
-ALTER BPOOL(BP38) VPSIZE(20000)
-ALTER BPOOL(BP39) VPSIZE(20000)
-ALTER BPOOL(BP40) VPSIZE(15000)
-ALTER BPOOL(BP41) VPSIZE(25000)
-ALTER BPOOL(BP42) VPSIZE(15000)
-ALTER BPOOL(BP43) VPSIZE(25000)
-ALTER BPOOL(BP44) VPSIZE(20000)
-ALTER BPOOL(BP45) VPSIZE(30000)
-ALTER BPOOL(BP46) VPSIZE(15000)
-ALTER BPOOL(BP47) VPSIZE(20000)
-ALTER BPOOL(BP48) VPSIZE(15000)
-ALTER BPOOL(BP49) VPSIZE(20000)
-ALTER BPOOL(BP16K0) VPSIZE(50000)
-ALTER BPOOL(BP16K1) VPSIZE(50000)
-ALTER BPOOL(BP16K4) VPSIZE(15000)
-ALTER BPOOL(BP16K8) VPSIZE(25000)
-ALTER BPOOL(BP16K9) VPSIZE(15000)
-ALTER BPOOL(BP32K1) VPSIZE(4000)
-ALTER BPOOL(BP7) VPSIZE(20000)
-ALTER BPOOL(BP32K) VPSIZE(4000)
```

Altering the Buffer Pool Thresholds

Modify buffer pool thresholds to correspond with the your stored object type.

```
-ALTER BPOOL(BP2)  VPSEQT(10) DWQT(50) VDWQT(10)
-ALTER BPOOL(BP10) VPSEQT(5) DWQT(80) VDWQT(5)
-ALTER BPOOL(BP12) VPSEQT(5) DWQT(80) VDWQT(5)
-ALTER BPOOL(BP14) VPSEQT(5) DWQT(80) VDWQT(5)
-ALTER BPOOL(BP16) VPSEQT(5) DWQT(80) VDWQT(5)
-ALTER BPOOL(BP17) VPSEQT(5) DWQT(80) VDWQT(5)
-ALTER BPOOL(BP18) VPSEQT(5) DWQT(80) VDWQT(5)
-ALTER BPOOL(BP22) VPSEQT(5) DWQT(80) VDWQT(80)
-ALTER BPOOL(BP23) VPSEQT(5) DWQT(80) VDWQT(80)
-ALTER BPOOL(BP25) VPSEQT(5) DWQT(80) VDWQT(80)
-ALTER BPOOL(BP26) VPSEQT(5) DWQT(80) VDWQT(80)
-ALTER BPOOL(BP27) VPSEQT(5) DWQT(80) VDWQT(80)
-ALTER BPOOL(BP30) VPSEQT(10) DWQT(50) VDWQT(10)
-ALTER BPOOL(BP32) VPSEQT(10) DWQT(50) VDWQT(10)
-ALTER BPOOL(BP33) VPSEQT(10) DWQT(50) VDWQT(10)
-ALTER BPOOL(BP34) VPSEQT(10) DWQT(50) VDWQT(10)
-ALTER BPOOL(BP35) VPSEQT(10) DWQT(50) VDWQT(10)
-ALTER BPOOL(BP36) VPSEQT(10) DWQT(50) VDWQT(10)
-ALTER BPOOL(BP37) VPSEQT(10) DWQT(50) VDWQT(10)
-ALTER BPOOL(BP38) VPSEQT(10) DWQT(50) VDWQT(10)
-ALTER BPOOL(BP39) VPSEQT(10) DWQT(50) VDWQT(10)
-ALTER BPOOL(BP40) VPSEQT(5) DWQT(80) VDWQT(80)
-ALTER BPOOL(BP41) VPSEQT(10) DWQT(50) VDWQT(10)
-ALTER BPOOL(BP42) VPSEQT(5) DWQT(80) VDWQT(80)
-ALTER BPOOL(BP43) VPSEQT(10) DWQT(50) VDWQT(10)
-ALTER BPOOL(BP44) VPSEQT(5) DWQT(80) VDWQT(80)
-ALTER BPOOL(BP45) VPSEQT(10) DWQT(50) VDWQT(10)
```

```
-ALTER BPOOL(BP46) VPSEQT(5) DWQT(80) VDWQT(80)

-ALTER BPOOL(BP47) VPSEQT(10) DWQT(50) VDWQT(10)

-ALTER BPOOL(BP48) VPSEQT(5) DWQT(80) VDWQT(80)

-ALTER BPOOL(BP49) VPSEQT(10) DWQT(50) VDWQT(10)

-ALTER BPOOL(BP16K0) VPSEQT(5) DWQT(80) VDWQT(80)

-ALTER BPOOL(BP16K1) VPSEQT(5) DWQT(80) VDWQT(80)

-ALTER BPOOL(BP16K4) VPSEQT(5) DWQT(80) VDWQT(80)

-ALTER BPOOL(BP16K8) VPSEQT(5) DWQT(80) VDWQT(80)

-ALTER BPOOL(BP16K9) VPSEQT(5) DWQT(80) VDWQT(80)

-ALTER BPOOL(BP32K1) VPSEQT(10) DWQT(50) VDWQT(10)
```

Altering the Objects

Alter the objects to correspond to the newly defined buffer pool settings.

Because the table space object names might not correspond to the table space in your environment, check these to make sure they are accurate. The following index examples relate to Siebel Financial Services.

```
S_ADDR_ORG

     ALTER TABLESPACE SIEB0008.AAAA489  BUFFERPOOL BP40;

     ALTER INDEX SIEBTO.S_ADDR_ORG_EI   BUFFERPOOL BP41;

     ALTER INDEX SIEBTO.S_ADDR_ORG_F1   BUFFERPOOL BP41;

     ALTER INDEX SIEBTO.S_ADDR_ORG_II   BUFFERPOOL BP41;

     ALTER INDEX SIEBTO.S_ADDR_ORG_M1   BUFFERPOOL BP41;

     ALTER INDEX SIEBTO.S_ADDR_ORG_M2   BUFFERPOOL BP41;

     ALTER INDEX SIEBTO.S_ADDR_ORG_M3   BUFFERPOOL BP41;

     ALTER INDEX SIEBTO.S_ADDR_ORG_M4   BUFFERPOOL BP41;

     ALTER INDEX SIEBTO.S_ADDR_ORG_M5   BUFFERPOOL BP41;

     ALTER INDEX SIEBTO.S_ADDR_ORG_M6   BUFFERPOOL BP41;

     ALTER INDEX SIEBTO.S_ADDR_ORG_P1   BUFFERPOOL BP41;

     ALTER INDEX SIEBTO.S_ADDR_ORG_P99  BUFFERPOOL BP41;

     ALTER INDEX SIEBTO.S_ADDR_ORG_U1   BUFFERPOOL BP41;
```

```
S_ADDR_PER

     ALTER TABLESPACE SIEB0008.AAAA494  BUFFERPOOL BP42;

     ALTER INDEX SIEBTO.S_ADDR_PER_EI   BUFFERPOOL BP43;

     ALTER INDEX SIEBTO.S_ADDR_PER_F50  BUFFERPOOL BP43;

     ALTER INDEX SIEBTO.S_ADDR_PER_II   BUFFERPOOL BP43;

     ALTER INDEX SIEBTO.S_ADDR_PER_M1   BUFFERPOOL BP43;

     ALTER INDEX SIEBTO.S_ADDR_PER_M2   BUFFERPOOL BP43;

     ALTER INDEX SIEBTO.S_ADDR_PER_M3   BUFFERPOOL BP43;

     ALTER INDEX SIEBTO.S_ADDR_PER_M4   BUFFERPOOL BP43;

     ALTER INDEX SIEBTO.S_ADDR_PER_M5   BUFFERPOOL BP43;

     ALTER INDEX SIEBTO.S_ADDR_PER_P1   BUFFERPOOL BP43;

     ALTER INDEX SIEBTO.S_ADDR_PER_P99  BUFFERPOOL BP43;

     ALTER INDEX SIEBTO.S_ADDR_PER_U1   BUFFERPOOL BP43;

S_ASSET

     ALTER TABLESPACE SIEB0008.AAAA585  BUFFERPOOL BP44;

     ALTER INDEX SIEBTO.S_ASSET_F10     BUFFERPOOL BP45;

     ALTER INDEX SIEBTO.S_ASSET_F11     BUFFERPOOL BP45;

     ALTER INDEX SIEBTO.S_ASSET_F12     BUFFERPOOL BP45;

     ALTER INDEX SIEBTO.S_ASSET_F13     BUFFERPOOL BP45;

     ALTER INDEX SIEBTO.S_ASSET_F14     BUFFERPOOL BP45;

     ALTER INDEX SIEBTO.S_ASSET_F2      BUFFERPOOL BP45;

     ALTER INDEX SIEBTO.S_ASSET_F4      BUFFERPOOL BP45;

     ALTER INDEX SIEBTO.S_ASSET_F5      BUFFERPOOL BP45;

     ALTER INDEX SIEBTO.S_ASSET_F50     BUFFERPOOL BP45;

     ALTER INDEX SIEBTO.S_ASSET_F51     BUFFERPOOL BP45;

     ALTER INDEX SIEBTO.S_ASSET_F52     BUFFERPOOL BP45;

     ALTER INDEX SIEBTO.S_ASSET_F53     BUFFERPOOL BP45;

     ALTER INDEX SIEBTO.S_ASSET_F54     BUFFERPOOL BP45;

     ALTER INDEX SIEBTO.S_ASSET_F6      BUFFERPOOL BP45;
     ALTER INDEX SIEBTO.S_ASSET_F7      BUFFERPOOL BP45;
     ALTER INDEX SIEBTO.S_ASSET_F8      BUFFERPOOL BP45;
     ALTER INDEX SIEBTO.S_ASSET_F9      BUFFERPOOL BP45;
     ALTER INDEX SIEBTO.S_ASSET_II      BUFFERPOOL BP45;
     ALTER INDEX SIEBTO.S_ASSET_M1      BUFFERPOOL BP45;
     ALTER INDEX SIEBTO.S_ASSET_M2      BUFFERPOOL BP45;
     ALTER INDEX SIEBTO.S_ASSET_M3      BUFFERPOOL BP45;
     ALTER INDEX SIEBTO.S_ASSET_M4      BUFFERPOOL BP45;
     ALTER INDEX SIEBTO.S_ASSET_M5      BUFFERPOOL BP45;
     ALTER INDEX SIEBTO.S_ASSET_M50     BUFFERPOOL BP45;
     ALTER INDEX SIEBTO.S_ASSET_M51     BUFFERPOOL BP45;
     ALTER INDEX SIEBTO.S_ASSET_M52     BUFFERPOOL BP45;
     ALTER INDEX SIEBTO.S_ASSET_M53     BUFFERPOOL BP45;
     ALTER INDEX SIEBTO.S_ASSET_M54     BUFFERPOOL BP45;
     ALTER INDEX SIEBTO.S_ASSET_P1      BUFFERPOOL BP45;
     ALTER INDEX SIEBTO.S_ASSET_P99     BUFFERPOOL BP45;
     ALTER INDEX SIEBTO.S_ASSET_U1      BUFFERPOOL BP45;
     ALTER INDEX SIEBTO.S_ASSET_U2      BUFFERPOOL BP45;
     ALTER INDEX SIEBTO.S_ASSET_U3      BUFFERPOOL BP45;
     ALTER INDEX SIEBTO.S_ASSET_U51     BUFFERPOOL BP45;
     ALTER INDEX SIEBTO.S_ASSET_V1      BUFFERPOOL BP45;
     ALTER INDEX SIEBTO.S_ASSET_V3      BUFFERPOOL BP45;
     ALTER INDEX SIEBTO.S_ASSET_V50     BUFFERPOOL BP45;
S_CONTACT
     ALTER TABLESPACE SIEB0012.BBBB826  BUFFERPOOL  BP16K0;
     ALTER INDEX SIEBTO.S_CONTACT_F1    BUFFERPOOL BP10;
     ALTER INDEX SIEBTO.S_CONTACT_F10   BUFFERPOOL BP10;
     ALTER INDEX SIEBTO.S_CONTACT_F11   BUFFERPOOL BP10;
     ALTER INDEX SIEBTO.S_CONTACT_F12   BUFFERPOOL BP10;
ALTER INDEX SIEBTO.S_CONTACT_F13   BUFFERPOOL BP10;

ALTER INDEX SIEBTO.S_CONTACT_F15   BUFFERPOOL BP10;

ALTER INDEX SIEBTO.S_CONTACT_F2    BUFFERPOOL BP10;

ALTER INDEX SIEBTO.S_CONTACT_F3    BUFFERPOOL BP10;

ALTER INDEX SIEBTO.S_CONTACT_F4    BUFFERPOOL BP10;

ALTER INDEX SIEBTO.S_CONTACT_F5    BUFFERPOOL BP10;

ALTER INDEX SIEBTO.S_CONTACT_F50   BUFFERPOOL BP10;

ALTER INDEX SIEBTO.S_CONTACT_F51   BUFFERPOOL BP10;

ALTER INDEX SIEBTO.S_CONTACT_F52   BUFFERPOOL BP10;

ALTER INDEX SIEBTO.S_CONTACT_F7    BUFFERPOOL BP10;

ALTER INDEX SIEBTO.S_CONTACT_II    BUFFERPOOL BP10;

ALTER INDEX SIEBTO.S_CONTACT_M1    BUFFERPOOL BP10;

ALTER INDEX SIEBTO.S_CONTACT_M12   BUFFERPOOL BP10;

ALTER INDEX SIEBTO.S_CONTACT_M18   BUFFERPOOL BP10;

ALTER INDEX SIEBTO.S_CONTACT_M19   BUFFERPOOL BP10;

ALTER INDEX SIEBTO.S_CONTACT_M2    BUFFERPOOL BP10;

ALTER INDEX SIEBTO.S_CONTACT_M20   BUFFERPOOL BP10;

ALTER INDEX SIEBTO.S_CONTACT_M21   BUFFERPOOL BP10;

ALTER INDEX SIEBTO.S_CONTACT_M22   BUFFERPOOL BP10;

ALTER INDEX SIEBTO.S_CONTACT_M3    BUFFERPOOL BP10;

ALTER INDEX SIEBTO.S_CONTACT_M4    BUFFERPOOL BP10;

ALTER INDEX SIEBTO.S_CONTACT_M50   BUFFERPOOL BP10;

ALTER INDEX SIEBTO.S_CONTACT_M51   BUFFERPOOL BP10;

ALTER INDEX SIEBTO.S_CONTACT_M6    BUFFERPOOL BP10;

ALTER INDEX SIEBTO.S_CONTACT_M8    BUFFERPOOL BP10;

ALTER INDEX SIEBTO.S_CONTACT_P1    BUFFERPOOL BP10;

ALTER INDEX SIEBTO.S_CONTACT_U2    BUFFERPOOL BP10;

ALTER INDEX SIEBTO.S_CONTACT_V1    BUFFERPOOL BP10;

ALTER INDEX SIEBTO.S_CONTACT_V2    BUFFERPOOL BP10;
```

```
     ALTER INDEX SIEBTO.S_CONTACT_V3    BUFFERPOOL BP10;
     ALTER INDEX SIEBTO.S_CONTACT_V5    BUFFERPOOL BP10;
     ALTER INDEX SIEBTO.S_CONTACT_EI    BUFFERPOOL BP30;
     ALTER INDEX SIEBTO.S_CONTACT_F6    BUFFERPOOL BP30;
     ALTER INDEX SIEBTO.S_CONTACT_M11   BUFFERPOOL BP30;
     ALTER INDEX SIEBTO.S_CONTACT_M13   BUFFERPOOL BP30;
     ALTER INDEX SIEBTO.S_CONTACT_M14   BUFFERPOOL BP30;
     ALTER INDEX SIEBTO.S_CONTACT_M15   BUFFERPOOL BP30;
     ALTER INDEX SIEBTO.S_CONTACT_M16   BUFFERPOOL BP30;
     ALTER INDEX SIEBTO.S_CONTACT_M17   BUFFERPOOL BP30;
     ALTER INDEX SIEBTO.S_CONTACT_M52   BUFFERPOOL BP30;
     ALTER INDEX SIEBTO.S_CONTACT_M53   BUFFERPOOL BP30;
     ALTER INDEX SIEBTO.S_CONTACT_M55   BUFFERPOOL BP30;
     ALTER INDEX SIEBTO.S_CONTACT_M56   BUFFERPOOL BP30;
     ALTER INDEX SIEBTO.S_CONTACT_M57   BUFFERPOOL BP30;
     ALTER INDEX SIEBTO.S_CONTACT_M58   BUFFERPOOL BP30;
     ALTER INDEX SIEBTO.S_CONTACT_M9    BUFFERPOOL BP30;
     ALTER INDEX SIEBTO.S_CONTACT_U1    BUFFERPOOL BP30;
S_DEFECT_PRDINT
     ALTER TABLESPACE SIEB0006.AAAA1053 BUFFERPOOL BP46;
     ALTER INDEX SIEBTO.S_DEFECT_PRDINT_U1 BUFFERPOOL BP47;
     ALTER INDEX SIEBTO.S_DEFECT_PRDINT_P1 BUFFERPOOL BP47;
     ALTER INDEX SIEBTO.S_DEFECT_PRDINT_F1 BUFFERPOOL BP47;
S_EVT_ACT
     ALTER TABLESPACE SIEB0009.BBBB1248 BUFFERPOOL BP16K9;
     ALTER INDEX SIEBTO.S_EVT_ACT_F1    BUFFERPOOL BP39;
     ALTER INDEX SIEBTO.S_EVT_ACT_F10   BUFFERPOOL BP39;
     ALTER INDEX SIEBTO.S_EVT_ACT_F11   BUFFERPOOL BP39;
     ALTER INDEX SIEBTO.S_EVT_ACT_F12   BUFFERPOOL BP39;
ALTER INDEX SIEBTO.S_EVT_ACT_F13   BUFFERPOOL BP39;
ALTER INDEX SIEBTO.S_EVT_ACT_F14   BUFFERPOOL BP39;
ALTER INDEX SIEBTO.S_EVT_ACT_F15   BUFFERPOOL BP39;
ALTER INDEX SIEBTO.S_EVT_ACT_F16   BUFFERPOOL BP39;
ALTER INDEX SIEBTO.S_EVT_ACT_F17   BUFFERPOOL BP39;
ALTER INDEX SIEBTO.S_EVT_ACT_F18   BUFFERPOOL BP39;
ALTER INDEX SIEBTO.S_EVT_ACT_F19   BUFFERPOOL BP39;
ALTER INDEX SIEBTO.S_EVT_ACT_F2    BUFFERPOOL BP39;
ALTER INDEX SIEBTO.S_EVT_ACT_F20   BUFFERPOOL BP39;
ALTER INDEX SIEBTO.S_EVT_ACT_F21   BUFFERPOOL BP39;
ALTER INDEX SIEBTO.S_EVT_ACT_F22   BUFFERPOOL BP39;
ALTER INDEX SIEBTO.S_EVT_ACT_F23   BUFFERPOOL BP39;
ALTER INDEX SIEBTO.S_EVT_ACT_F24   BUFFERPOOL BP39;
ALTER INDEX SIEBTO.S_EVT_ACT_F25   BUFFERPOOL BP39;
ALTER INDEX SIEBTO.S_EVT_ACT_F27   BUFFERPOOL BP39;
ALTER INDEX SIEBTO.S_EVT_ACT_F28   BUFFERPOOL BP39;
ALTER INDEX SIEBTO.S_EVT_ACT_F29   BUFFERPOOL BP39;
ALTER INDEX SIEBTO.S_EVT_ACT_F3    BUFFERPOOL BP39;
ALTER INDEX SIEBTO.S_EVT_ACT_F30   BUFFERPOOL BP39;
ALTER INDEX SIEBTO.S_EVT_ACT_F31   BUFFERPOOL BP39;
ALTER INDEX SIEBTO.S_EVT_ACT_F32   BUFFERPOOL BP39;
ALTER INDEX SIEBTO.S_EVT_ACT_F33   BUFFERPOOL BP39;
ALTER INDEX SIEBTO.S_EVT_ACT_F34   BUFFERPOOL BP39;
ALTER INDEX SIEBTO.S_EVT_ACT_F35   BUFFERPOOL BP39;
ALTER INDEX SIEBTO.S_EVT_ACT_F36   BUFFERPOOL BP39;
ALTER INDEX SIEBTO.S_EVT_ACT_F37   BUFFERPOOL BP39;
ALTER INDEX SIEBTO.S_EVT_ACT_F38   BUFFERPOOL BP39;
ALTER INDEX SIEBTO.S_EVT_ACT_F5    BUFFERPOOL BP39;
ALTER INDEX SIEBTO.S_EVT_ACT_F50   BUFFERPOOL BP39;
ALTER INDEX SIEBTO.S_EVT_ACT_F51   BUFFERPOOL BP39;
ALTER INDEX SIEBTO.S_EVT_ACT_F52   BUFFERPOOL BP39;
ALTER INDEX SIEBTO.S_EVT_ACT_F53   BUFFERPOOL BP39;
ALTER INDEX SIEBTO.S_EVT_ACT_F54   BUFFERPOOL BP39;
ALTER INDEX SIEBTO.S_EVT_ACT_F55   BUFFERPOOL BP39;
ALTER INDEX SIEBTO.S_EVT_ACT_F56   BUFFERPOOL BP39;
ALTER INDEX SIEBTO.S_EVT_ACT_F57   BUFFERPOOL BP39;
ALTER INDEX SIEBTO.S_EVT_ACT_F58   BUFFERPOOL BP39;
ALTER INDEX SIEBTO.S_EVT_ACT_F59   BUFFERPOOL BP39;
ALTER INDEX SIEBTO.S_EVT_ACT_F6    BUFFERPOOL BP39;
ALTER INDEX SIEBTO.S_EVT_ACT_F60   BUFFERPOOL BP39;
ALTER INDEX SIEBTO.S_EVT_ACT_F61   BUFFERPOOL BP39;
ALTER INDEX SIEBTO.S_EVT_ACT_F62   BUFFERPOOL BP39;
ALTER INDEX SIEBTO.S_EVT_ACT_F7    BUFFERPOOL BP39;
ALTER INDEX SIEBTO.S_EVT_ACT_F8    BUFFERPOOL BP39;
ALTER INDEX SIEBTO.S_EVT_ACT_F9    BUFFERPOOL BP39;
ALTER INDEX SIEBTO.S_EVT_ACT_M1    BUFFERPOOL BP39;
ALTER INDEX SIEBTO.S_EVT_ACT_M10   BUFFERPOOL BP39;
ALTER INDEX SIEBTO.S_EVT_ACT_M11   BUFFERPOOL BP39;
ALTER INDEX SIEBTO.S_EVT_ACT_M2    BUFFERPOOL BP39;
ALTER INDEX SIEBTO.S_EVT_ACT_M3    BUFFERPOOL BP39;
ALTER INDEX SIEBTO.S_EVT_ACT_M5    BUFFERPOOL BP39;
ALTER INDEX SIEBTO.S_EVT_ACT_M50   BUFFERPOOL BP39;
ALTER INDEX SIEBTO.S_EVT_ACT_M51   BUFFERPOOL BP39;
ALTER INDEX SIEBTO.S_EVT_ACT_M6    BUFFERPOOL BP39;
ALTER INDEX SIEBTO.S_EVT_ACT_M7    BUFFERPOOL BP39;
ALTER INDEX SIEBTO.S_EVT_ACT_M8    BUFFERPOOL BP39;
ALTER INDEX SIEBTO.S_EVT_ACT_M9    BUFFERPOOL BP39;
ALTER INDEX SIEBTO.S_EVT_ACT_P1    BUFFERPOOL BP39;
```

```
     ALTER INDEX SIEBTO.S_EVT_ACT_P99   BUFFERPOOL BP39;
     ALTER INDEX SIEBTO.S_EVT_ACT_U1    BUFFERPOOL BP39;
     ALTER INDEX SIEBTO.S_EVT_ACT_V1    BUFFERPOOL BP39;
     ALTER INDEX SIEBTO.S_EVT_ACT_V2    BUFFERPOOL BP39;
S_INS_ITEM
     ALTER TABLESPACE SIEB0015.AAAA1513 BUFFERPOOL BP48;
     ALTER INDEX SIEBTO.S_INS_ITEM_F1   BUFFERPOOL BP49;
     ALTER INDEX SIEBTO.S_INS_ITEM_F3   BUFFERPOOL BP49;
     ALTER INDEX SIEBTO.S_INS_ITEM_F4   BUFFERPOOL BP49;
     ALTER INDEX SIEBTO.S_INS_ITEM_F5   BUFFERPOOL BP49;
     ALTER INDEX SIEBTO.S_INS_ITEM_F50  BUFFERPOOL BP49;
     ALTER INDEX SIEBTO.S_INS_ITEM_M1   BUFFERPOOL BP49;
     ALTER INDEX SIEBTO.S_INS_ITEM_P1   BUFFERPOOL BP49;
     ALTER INDEX SIEBTO.S_INS_ITEM_P99  BUFFERPOOL BP49;
     ALTER INDEX SIEBTO.S_INS_ITEM_U1   BUFFERPOOL BP49;
S_OPTY
     ALTER TABLESPACE SIEB0010.AAAA1689 BUFFERPOOL BP22;
     ALTER INDEX SIEBTO.S_OPTY_F1       BUFFERPOOL BP12;
     ALTER INDEX SIEBTO.S_OPTY_F3       BUFFERPOOL BP12;
     ALTER INDEX SIEBTO.S_OPTY_F4       BUFFERPOOL BP12;
     ALTER INDEX SIEBTO.S_OPTY_F5       BUFFERPOOL BP12;
     ALTER INDEX SIEBTO.S_OPTY_F50      BUFFERPOOL BP12;
     ALTER INDEX SIEBTO.S_OPTY_F51      BUFFERPOOL BP12;
     ALTER INDEX SIEBTO.S_OPTY_F52      BUFFERPOOL BP12;
     ALTER INDEX SIEBTO.S_OPTY_F53      BUFFERPOOL BP12;
     ALTER INDEX SIEBTO.S_OPTY_F54      BUFFERPOOL BP12;
     ALTER INDEX SIEBTO.S_OPTY_F55      BUFFERPOOL BP12;
     ALTER INDEX SIEBTO.S_OPTY_F56      BUFFERPOOL BP12;
     ALTER INDEX SIEBTO.S_OPTY_F57      BUFFERPOOL BP12;
ALTER INDEX SIEBTO.S_OPTY_F58      BUFFERPOOL BP12;
ALTER INDEX SIEBTO.S_OPTY_F6       BUFFERPOOL BP12;
ALTER INDEX SIEBTO.S_OPTY_F7       BUFFERPOOL BP12;
ALTER INDEX SIEBTO.S_OPTY_F8       BUFFERPOOL BP12;
ALTER INDEX SIEBTO.S_OPTY_II       BUFFERPOOL BP12;
ALTER INDEX SIEBTO.S_OPTY_M1       BUFFERPOOL BP12;
ALTER INDEX SIEBTO.S_OPTY_M2       BUFFERPOOL BP12;
ALTER INDEX SIEBTO.S_OPTY_M3       BUFFERPOOL BP12;
ALTER INDEX SIEBTO.S_OPTY_M4       BUFFERPOOL BP12;
ALTER INDEX SIEBTO.S_OPTY_M50      BUFFERPOOL BP12;
ALTER INDEX SIEBTO.S_OPTY_M51      BUFFERPOOL BP12;
ALTER INDEX SIEBTO.S_OPTY_M52      BUFFERPOOL BP12;
ALTER INDEX SIEBTO.S_OPTY_M53      BUFFERPOOL BP12;
ALTER INDEX SIEBTO.S_OPTY_M54      BUFFERPOOL BP12;
ALTER INDEX SIEBTO.S_OPTY_M55      BUFFERPOOL BP12;
ALTER INDEX SIEBTO.S_OPTY_M56      BUFFERPOOL BP12;
ALTER INDEX SIEBTO.S_OPTY_M57      BUFFERPOOL BP12;
ALTER INDEX SIEBTO.S_OPTY_M6       BUFFERPOOL BP12;
ALTER INDEX SIEBTO.S_OPTY_M8       BUFFERPOOL BP12;
ALTER INDEX SIEBTO.S_OPTY_M9       BUFFERPOOL BP12;
ALTER INDEX SIEBTO.S_OPTY_P1       BUFFERPOOL BP12;
ALTER INDEX SIEBTO.S_OPTY_U1       BUFFERPOOL BP32;
ALTER INDEX SIEBTO.S_OPTY_U2       BUFFERPOOL BP32;
ALTER INDEX SIEBTO.S_OPTY_V1       BUFFERPOOL BP32;
ALTER INDEX SIEBTO.S_OPTY_V2       BUFFERPOOL BP32;
ALTER INDEX SIEBTO.S_OPTY_V3       BUFFERPOOL BP32;
ALTER INDEX SIEBTO.S_OPTY_V4       BUFFERPOOL BP32;
ALTER INDEX SIEBTO.S_OPTY_V5       BUFFERPOOL BP32;
```

```
S_OPTY_BU
     ALTER TABLESPACE SIEB0006.AAAA1701 BUFFERPOOL BP23;
     ALTER INDEX SIEBTO.S_OPTY_BU_M1    BUFFERPOOL BP33;
     ALTER INDEX SIEBTO.S_OPTY_BU_M2    BUFFERPOOL BP33;
     ALTER INDEX SIEBTO.S_OPTY_BU_M3    BUFFERPOOL BP33;
     ALTER INDEX SIEBTO.S_OPTY_BU_M4    BUFFERPOOL BP33;
     ALTER INDEX SIEBTO.S_OPTY_BU_M5    BUFFERPOOL BP33;
     ALTER INDEX SIEBTO.S_OPTY_BU_M6    BUFFERPOOL BP33;
     ALTER INDEX SIEBTO.S_OPTY_BU_M7    BUFFERPOOL BP33;
     ALTER INDEX SIEBTO.S_OPTY_BU_M8    BUFFERPOOL BP33;
     ALTER INDEX SIEBTO.S_OPTY_BU_M9    BUFFERPOOL BP33;
     ALTER INDEX SIEBTO.S_OPTY_BU_P1    BUFFERPOOL BP33;
     ALTER INDEX SIEBTO.S_OPTY_BU_U1    BUFFERPOOL BP33;
S_ORG_EXT
     ALTER TABLESPACE SIEB0012.BBBB1792 BUFFERPOOL BP16K4;
     ALTER INDEX SIEBTO.S_ORG_EXT_F1    BUFFERPOOL BP14;
     ALTER INDEX SIEBTO.S_ORG_EXT_F10   BUFFERPOOL BP14;
     ALTER INDEX SIEBTO.S_ORG_EXT_F11   BUFFERPOOL BP14;
     ALTER INDEX SIEBTO.S_ORG_EXT_F12   BUFFERPOOL BP14;
     ALTER INDEX SIEBTO.S_ORG_EXT_F13   BUFFERPOOL BP14;
     ALTER INDEX SIEBTO.S_ORG_EXT_F14   BUFFERPOOL BP14;
     ALTER INDEX SIEBTO.S_ORG_EXT_F15   BUFFERPOOL BP14;
     ALTER INDEX SIEBTO.S_ORG_EXT_F16   BUFFERPOOL BP14;
     ALTER INDEX SIEBTO.S_ORG_EXT_F17   BUFFERPOOL BP14;
     ALTER INDEX SIEBTO.S_ORG_EXT_F18   BUFFERPOOL BP14;
     ALTER INDEX SIEBTO.S_ORG_EXT_F19   BUFFERPOOL BP14;
     ALTER INDEX SIEBTO.S_ORG_EXT_F2    BUFFERPOOL BP14;
     ALTER INDEX SIEBTO.S_ORG_EXT_F20   BUFFERPOOL BP14;
     ALTER INDEX SIEBTO.S_ORG_EXT_F3    BUFFERPOOL BP14;
ALTER INDEX SIEBTO.S_ORG_EXT_F4    BUFFERPOOL BP14;
ALTER INDEX SIEBTO.S_ORG_EXT_F5    BUFFERPOOL BP14;
ALTER INDEX SIEBTO.S_ORG_EXT_F50   BUFFERPOOL BP14;
ALTER INDEX SIEBTO.S_ORG_EXT_F51   BUFFERPOOL BP14;
ALTER INDEX SIEBTO.S_ORG_EXT_F52   BUFFERPOOL BP14;
ALTER INDEX SIEBTO.S_ORG_EXT_F53   BUFFERPOOL BP14;
ALTER INDEX SIEBTO.S_ORG_EXT_F54   BUFFERPOOL BP14;
ALTER INDEX SIEBTO.S_ORG_EXT_F55   BUFFERPOOL BP14;
ALTER INDEX SIEBTO.S_ORG_EXT_F56   BUFFERPOOL BP14;
ALTER INDEX SIEBTO.S_ORG_EXT_F57   BUFFERPOOL BP14;
ALTER INDEX SIEBTO.S_ORG_EXT_F58   BUFFERPOOL BP14;
ALTER INDEX SIEBTO.S_ORG_EXT_F59   BUFFERPOOL BP14;
ALTER INDEX SIEBTO.S_ORG_EXT_F6    BUFFERPOOL BP14;
ALTER INDEX SIEBTO.S_ORG_EXT_F60   BUFFERPOOL BP14;
ALTER INDEX SIEBTO.S_ORG_EXT_F61   BUFFERPOOL BP14;
ALTER INDEX SIEBTO.S_ORG_EXT_F62   BUFFERPOOL BP14;
ALTER INDEX SIEBTO.S_ORG_EXT_F7    BUFFERPOOL BP14;
ALTER INDEX SIEBTO.S_ORG_EXT_F8    BUFFERPOOL BP14;
ALTER INDEX SIEBTO.S_ORG_EXT_F9    BUFFERPOOL BP14;
ALTER INDEX SIEBTO.S_ORG_EXT_II    BUFFERPOOL BP14;
ALTER INDEX SIEBTO.S_ORG_EXT_M1    BUFFERPOOL BP14;
ALTER INDEX SIEBTO.S_ORG_EXT_M10   BUFFERPOOL BP14;
ALTER INDEX SIEBTO.S_ORG_EXT_M12   BUFFERPOOL BP14;
ALTER INDEX SIEBTO.S_ORG_EXT_M13   BUFFERPOOL BP14;
ALTER INDEX SIEBTO.S_ORG_EXT_M14   BUFFERPOOL BP14;
ALTER INDEX SIEBTO.S_ORG_EXT_M19   BUFFERPOOL BP14;
ALTER INDEX SIEBTO.S_ORG_EXT_M3    BUFFERPOOL BP14;
ALTER INDEX SIEBTO.S_ORG_EXT_M4    BUFFERPOOL BP14;
ALTER INDEX SIEBTO.S_ORG_EXT_M50   BUFFERPOOL BP14;
```

```
     ALTER INDEX SIEBTO.S_ORG_EXT_M51   BUFFERPOOL BP14;
     ALTER INDEX SIEBTO.S_ORG_EXT_M52   BUFFERPOOL BP14;
     ALTER INDEX SIEBTO.S_ORG_EXT_M53   BUFFERPOOL BP14;
     ALTER INDEX SIEBTO.S_ORG_EXT_M6    BUFFERPOOL BP14;
     ALTER INDEX SIEBTO.S_ORG_EXT_M7    BUFFERPOOL BP14;
     ALTER INDEX SIEBTO.S_ORG_EXT_M8    BUFFERPOOL BP14;
     ALTER INDEX SIEBTO.S_ORG_EXT_P1    BUFFERPOOL BP14;
     ALTER INDEX SIEBTO.S_ORG_EXT_U3    BUFFERPOOL BP14;
     ALTER INDEX SIEBTO.S_ORG_EXT_V1    BUFFERPOOL BP14;
     ALTER INDEX SIEBTO.S_ORG_EXT_V2    BUFFERPOOL BP14;
     ALTER INDEX SIEBTO.S_ORG_EXT_V3    BUFFERPOOL BP14;
     ALTER INDEX SIEBTO.S_ORG_EXT_V4    BUFFERPOOL BP14;
     ALTER INDEX SIEBTO.S_ORG_EXT_V5    BUFFERPOOL BP14;
     ALTER INDEX SIEBTO.S_ORG_EXT_V6    BUFFERPOOL BP14;
     ALTER INDEX SIEBTO.S_ORG_EXT_V7    BUFFERPOOL BP14;
     ALTER INDEX SIEBTO.S_ORG_EXT_V8    BUFFERPOOL BP14;
     ALTER INDEX SIEBTO.S_ORG_EXT_V9    BUFFERPOOL BP14;
     ALTER INDEX SIEBTO.S_ORG_EXT_M11   BUFFERPOOL BP34;
     ALTER INDEX SIEBTO.S_ORG_EXT_M15   BUFFERPOOL BP34;
     ALTER INDEX SIEBTO.S_ORG_EXT_M16   BUFFERPOOL BP34;
     ALTER INDEX SIEBTO.S_ORG_EXT_M17   BUFFERPOOL BP34;
     ALTER INDEX SIEBTO.S_ORG_EXT_M2    BUFFERPOOL BP34;
     ALTER INDEX SIEBTO.S_ORG_EXT_M20   BUFFERPOOL BP34;
     ALTER INDEX SIEBTO.S_ORG_EXT_M5    BUFFERPOOL BP34;
     ALTER INDEX SIEBTO.S_ORG_EXT_M9    BUFFERPOOL BP34;
     ALTER INDEX SIEBTO.S_ORG_EXT_U1    BUFFERPOOL BP34;
     ALTER INDEX SIEBTO.S_ORG_EXT_U2    BUFFERPOOL BP34;
S_ORG_GROUP
     ALTER TABLESPACE SIEB0012.AAAA1807 BUFFERPOOL BP25;
     ALTER INDEX SIEBTO.S_ORG_GROUP_F1  BUFFERPOOL BP35;
     ALTER INDEX SIEBTO.S_ORG_GROUP_F2  BUFFERPOOL BP35;

     ALTER INDEX SIEBTO.S_ORG_GROUP_F50 BUFFERPOOL BP35;

     ALTER INDEX SIEBTO.S_ORG_GROUP_M1  BUFFERPOOL BP35;

     ALTER INDEX SIEBTO.S_ORG_GROUP_M50 BUFFERPOOL BP35;

     ALTER INDEX SIEBTO.S_ORG_GROUP_P1  BUFFERPOOL BP35;

     ALTER INDEX SIEBTO.S_ORG_GROUP_U1  BUFFERPOOL BP35;

     ALTER INDEX SIEBTO.S_ORG_GROUP_U2  BUFFERPOOL BP35;

     ALTER INDEX SIEBTO.S_ORG_GROUP_V1  BUFFERPOOL BP35;

     ALTER INDEX SIEBTO.S_ORG_GROUP_V2  BUFFERPOOL BP35;

S_PARTY

     ALTER TABLESPACE SIEB0010.AAAA1853 BUFFERPOOL BP26;

     ALTER INDEX SIEBTO.S_PARTY_F1      BUFFERPOOL BP16;

     ALTER INDEX SIEBTO.S_PARTY_M1      BUFFERPOOL BP16;

     ALTER INDEX SIEBTO.S_PARTY_M2      BUFFERPOOL BP16;

     ALTER INDEX SIEBTO.S_PARTY_M3      BUFFERPOOL BP16;

     ALTER INDEX SIEBTO.S_PARTY_M4      BUFFERPOOL BP16;

     ALTER INDEX SIEBTO.S_PARTY_P1      BUFFERPOOL BP16;

     ALTER INDEX SIEBTO.S_PARTY_P99     BUFFERPOOL BP16;

     ALTER INDEX SIEBTO.S_PARTY_U1      BUFFERPOOL BP36;

S_PROD_INT

     ALTER TABLESPACE SIEB0010.AAAA1992 BUFFERPOOL BP27;

     ALTER INDEX SIEBTO.S_PROD_INT_F10  BUFFERPOOL BP17;

     ALTER INDEX SIEBTO.S_PROD_INT_F11  BUFFERPOOL BP17;

     ALTER INDEX SIEBTO.S_PROD_INT_F12  BUFFERPOOL BP17;

     ALTER INDEX SIEBTO.S_PROD_INT_F14  BUFFERPOOL BP17;

     ALTER INDEX SIEBTO.S_PROD_INT_F2   BUFFERPOOL BP17;

     ALTER INDEX SIEBTO.S_PROD_INT_F3   BUFFERPOOL BP17;

     ALTER INDEX SIEBTO.S_PROD_INT_F4   BUFFERPOOL BP17;

     ALTER INDEX SIEBTO.S_PROD_INT_F5   BUFFERPOOL BP17;
ALTER INDEX SIEBTO.S_PROD_INT_F50  BUFFERPOOL BP17;
ALTER INDEX SIEBTO.S_PROD_INT_F51  BUFFERPOOL BP17;
ALTER INDEX SIEBTO.S_PROD_INT_F52  BUFFERPOOL BP17;
ALTER INDEX SIEBTO.S_PROD_INT_F53  BUFFERPOOL BP17;
ALTER INDEX SIEBTO.S_PROD_INT_F6   BUFFERPOOL BP17;
ALTER INDEX SIEBTO.S_PROD_INT_F7   BUFFERPOOL BP17;
ALTER INDEX SIEBTO.S_PROD_INT_F8   BUFFERPOOL BP17;
ALTER INDEX SIEBTO.S_PROD_INT_F9   BUFFERPOOL BP17;
ALTER INDEX SIEBTO.S_PROD_INT_II   BUFFERPOOL BP17;
ALTER INDEX SIEBTO.S_PROD_INT_M1   BUFFERPOOL BP17;
ALTER INDEX SIEBTO.S_PROD_INT_M10  BUFFERPOOL BP17;
ALTER INDEX SIEBTO.S_PROD_INT_M11  BUFFERPOOL BP17;
ALTER INDEX SIEBTO.S_PROD_INT_M14  BUFFERPOOL BP17;
ALTER INDEX SIEBTO.S_PROD_INT_M15  BUFFERPOOL BP17;
ALTER INDEX SIEBTO.S_PROD_INT_M16  BUFFERPOOL BP17;
ALTER INDEX SIEBTO.S_PROD_INT_M4   BUFFERPOOL BP17;
ALTER INDEX SIEBTO.S_PROD_INT_M9   BUFFERPOOL BP17;
ALTER INDEX SIEBTO.S_PROD_INT_P1   BUFFERPOOL BP17;
ALTER INDEX SIEBTO.S_PROD_INT_V1   BUFFERPOOL BP17;
ALTER INDEX SIEBTO.S_PROD_INT_V2   BUFFERPOOL BP17;
ALTER INDEX SIEBTO.S_PROD_INT_V3   BUFFERPOOL BP17;
ALTER INDEX SIEBTO.S_PROD_INT_M12  BUFFERPOOL BP37;
ALTER INDEX SIEBTO.S_PROD_INT_M13  BUFFERPOOL BP37;
ALTER INDEX SIEBTO.S_PROD_INT_M2   BUFFERPOOL BP37;
ALTER INDEX SIEBTO.S_PROD_INT_M3   BUFFERPOOL BP37;
ALTER INDEX SIEBTO.S_PROD_INT_M5   BUFFERPOOL BP37;
ALTER INDEX SIEBTO.S_PROD_INT_M50  BUFFERPOOL BP37;
ALTER INDEX SIEBTO.S_PROD_INT_M51  BUFFERPOOL BP37;
ALTER INDEX SIEBTO.S_PROD_INT_M6   BUFFERPOOL BP37;
```

```
     ALTER INDEX SIEBTO.S_PROD_INT_M7   BUFFERPOOL BP37;
     ALTER INDEX SIEBTO.S_PROD_INT_M8   BUFFERPOOL BP37;
     ALTER INDEX SIEBTO.S_PROD_INT_U1   BUFFERPOOL BP37;
     ALTER INDEX SIEBTO.S_PROD_INT_U2   BUFFERPOOL BP37;
     ALTER INDEX SIEBTO.S_PROD_INT_U3   BUFFERPOOL BP37;
S_SRV_REQ
     ALTER TABLESPACE SIEB0011.BBBB2291 BUFFERPOOL BP16K8;
     ALTER INDEX SIEBTO.S_SRV_REQ_F1    BUFFERPOOL BP18;
     ALTER INDEX SIEBTO.S_SRV_REQ_F10   BUFFERPOOL BP18;
     ALTER INDEX SIEBTO.S_SRV_REQ_F12   BUFFERPOOL BP18;
     ALTER INDEX SIEBTO.S_SRV_REQ_F13   BUFFERPOOL BP18;
     ALTER INDEX SIEBTO.S_SRV_REQ_F14   BUFFERPOOL BP18;
     ALTER INDEX SIEBTO.S_SRV_REQ_F15   BUFFERPOOL BP18;
     ALTER INDEX SIEBTO.S_SRV_REQ_F16   BUFFERPOOL BP18;
     ALTER INDEX SIEBTO.S_SRV_REQ_F17   BUFFERPOOL BP18;
     ALTER INDEX SIEBTO.S_SRV_REQ_F18   BUFFERPOOL BP18;
     ALTER INDEX SIEBTO.S_SRV_REQ_F19   BUFFERPOOL BP18;
     ALTER INDEX SIEBTO.S_SRV_REQ_F2    BUFFERPOOL BP18;
     ALTER INDEX SIEBTO.S_SRV_REQ_F20   BUFFERPOOL BP18;
     ALTER INDEX SIEBTO.S_SRV_REQ_F21   BUFFERPOOL BP18;
     ALTER INDEX SIEBTO.S_SRV_REQ_F22   BUFFERPOOL BP18;
     ALTER INDEX SIEBTO.S_SRV_REQ_F23   BUFFERPOOL BP18;
     ALTER INDEX SIEBTO.S_SRV_REQ_F24   BUFFERPOOL BP18;
     ALTER INDEX SIEBTO.S_SRV_REQ_F3    BUFFERPOOL BP18;
     ALTER INDEX SIEBTO.S_SRV_REQ_F4    BUFFERPOOL BP18;
     ALTER INDEX SIEBTO.S_SRV_REQ_F5    BUFFERPOOL BP18;
     ALTER INDEX SIEBTO.S_SRV_REQ_F50   BUFFERPOOL BP18;
     ALTER INDEX SIEBTO.S_SRV_REQ_F51   BUFFERPOOL BP18;
     ALTER INDEX SIEBTO.S_SRV_REQ_F52  BUFFERPOOL BP18;
ALTER INDEX SIEBTO.S_SRV_REQ_F53  BUFFERPOOL BP18;

ALTER INDEX SIEBTO.S_SRV_REQ_F54  BUFFERPOOL BP18;

ALTER INDEX SIEBTO.S_SRV_REQ_F55  BUFFERPOOL BP18;

ALTER INDEX SIEBTO.S_SRV_REQ_F56   BUFFERPOOL BP18;

ALTER INDEX SIEBTO.S_SRV_REQ_F57   BUFFERPOOL BP18;

ALTER INDEX SIEBTO.S_SRV_REQ_F58   BUFFERPOOL BP18;

ALTER INDEX SIEBTO.S_SRV_REQ_F59   BUFFERPOOL BP18;

ALTER INDEX SIEBTO.S_SRV_REQ_F6    BUFFERPOOL BP18;

ALTER INDEX SIEBTO.S_SRV_REQ_F7    BUFFERPOOL BP18;

ALTER INDEX SIEBTO.S_SRV_REQ_F9    BUFFERPOOL BP18;

ALTER INDEX SIEBTO.S_SRV_REQ_M10   BUFFERPOOL BP18;

ALTER INDEX SIEBTO.S_SRV_REQ_M2    BUFFERPOOL BP18;

ALTER INDEX SIEBTO.S_SRV_REQ_M3    BUFFERPOOL BP18;

ALTER INDEX SIEBTO.S_SRV_REQ_M4    BUFFERPOOL BP18;

ALTER INDEX SIEBTO.S_SRV_REQ_M5    BUFFERPOOL BP18;

ALTER INDEX SIEBTO.S_SRV_REQ_M50   BUFFERPOOL BP18;

ALTER INDEX SIEBTO.S_SRV_REQ_M51   BUFFERPOOL BP18;

ALTER INDEX SIEBTO.S_SRV_REQ_M6    BUFFERPOOL BP18;

ALTER INDEX SIEBTO.S_SRV_REQ_M9    BUFFERPOOL BP18;

ALTER INDEX SIEBTO.S_SRV_REQ_P99   BUFFERPOOL BP18;

ALTER INDEX SIEBTO.S_SRV_REQ_M1    BUFFERPOOL BP38;

ALTER INDEX SIEBTO.S_SRV_REQ_M7    BUFFERPOOL BP38;

ALTER INDEX SIEBTO.S_SRV_REQ_M8    BUFFERPOOL BP38;

ALTER INDEX SIEBTO.S_SRV_REQ_P1    BUFFERPOOL BP38;

ALTER INDEX SIEBTO.S_SRV_REQ_U1    BUFFERPOOL BP38;

ALTER INDEX SIEBTO.S_SRV_REQ_U2    BUFFERPOOL BP38;
```

Stopping and Restarting the Databases

You must stop and restart the databases to make sure that the new buffer pools have been allocated.

NOTE: When taking advantage of DB2 data sharing, you must stop objects before altering them.

To stop the database

- Enter the following commands:

```
-STO DB(SIEB0001) SPACENAM(*)

-STO DB(SIEB0002) SPACENAM(*)

-STO DB(SIEB0003) SPACENAM(*)

-STO DB(SIEB0004) SPACENAM(*)

-STO DB(SIEB0005) SPACENAM(*)

-STO DB(SIEB0006) SPACENAM(*)

-STO DB(SIEB0007) SPACENAM(*)

-STO DB(SIEB0008) SPACENAM(*)

-STO DB(SIEB0009) SPACENAM(*)

-STO DB(SIEB0010) SPACENAM(*)

-STO DB(SIEB0011) SPACENAM(*)

-STO DB(SIEB0012) SPACENAM(*)

-STO DB(SIEB0013) SPACENAM(*)

-STO DB(SIEB0014) SPACENAM(*)

-STO DB(SIEB0015) SPACENAM(*)

-STO DB(SIEB0016) SPACENAM(*)
```

To restart the databases

- Enter the following commands:

```
-STA DB(SIEB0001) SPACENAM(*)

-STA DB(SIEB0002) SPACENAM(*)
```

```
-STA DB(SIEB0003) SPACENAM(*)

-STA DB(SIEB0004) SPACENAM(*)

-STA DB(SIEB0005) SPACENAM(*)

-STA DB(SIEB0006) SPACENAM(*)

-STA DB(SIEB0007) SPACENAM(*)

-STA DB(SIEB0008) SPACENAM(*)

-STA DB(SIEB0009) SPACENAM(*)

-STA DB(SIEB0010) SPACENAM(*)

-STA DB(SIEB0011) SPACENAM(*)

-STA DB(SIEB0012) SPACENAM(*)

-STA DB(SIEB0013) SPACENAM(*)

-STA DB(SIEB0014) SPACENAM(*)

-STA DB(SIEB0015) SPACENAM(*)

-STA DB(SIEB0016) SPACENAM(*)
```

Index