Migrating Siebel Database from DB2/Oracle for NT to DB2 for OS/390

November 2001
Take Note! Before using this information and the product it supports, be sure to read the general information in “Special notices” on page 209.

First Edition (November 2001)

This edition applies to Siebel 2000 Version 6.2.1 for use with Oracle for NT V8.1.6, DB2 for NT V6 and DB2 for OS/390 V6.

Comments may be addressed to:
IBM Corporation, International Technical Support Organization
Dept. HYJ Mail Station P099
2455 South Road
Poughkeepsie, NY 12601-5400

When you send information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© Copyright International Business Machines Corporation 2001. All rights reserved.
Note to U.S Government Users – Documentation related to restricted rights – Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.
## Contents

Preface .............................................................. ix
The team that wrote this redbook ................................. ix
Special notice ......................................................... xi
IBM Trademarks ....................................................... xii
Comments welcome .................................................. xii

### Chapter 1. Introducing Siebel database migration...

1.1 Siebel architecture and configurations .................. 2
1.2 When should you migrate the Siebel database .......... 2
1.3 Siebel database migration ..................................... 3
1.4 Siebel database structure ..................................... 4
   1.4.1 The repository .............................................. 4
   1.4.2 The data ..................................................... 5
1.5 Which tool should you use to move data ................. 6
   1.5.1 Siebel database utilities ................................. 6
   1.5.2 RDBMS-specific database utilities .................. 7
   1.5.3 Data warehousing tools ................................. 7
1.6 Methodology for migrating to DB2 for OS/390 .......... 7
1.7 Migration scenarios .......................................... 8

### Chapter 2. Implementing the target Siebel environment on OS/390...

2.1 Preparing the OS/390 environment for the Siebel database 13
   2.1.1 Checking the maintenance level ...................... 13
   2.1.2 Creating RACF user IDs .................................. 13
   2.1.3 Creating a large VTOC .................................... 14
   2.1.4 Creating DB2 storage groups ......................... 14
   2.1.5 Customizing DB2 buffer pools ....................... 14
   2.1.6 Customizing the EDM pool ............................. 14
   2.1.7 Customizing DSNZPARM parameters ................ 14
   2.1.8 Enabling DB2 stored procedures ................... 15
2.2 Enabling database connections to DB2 for OS/390 .......... 16
   2.2.1 DDF setup on OS/390 ..................................... 16
   2.2.2 DB2 Connect EE setup on NT .......................... 16
   2.2.3 Client setup on a PC .................................... 17
2.3 Installing Siebel 2000 on the NT server .................. 18
   2.3.1 Creating required NT user accounts ................ 18
   2.3.2 Creating the Siebel File System .................... 18
   2.3.3 Installing the Siebel Gateway Server ............. 18

© Copyright IBM Corp. 2001
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Appendix A. Implementing the Siebel source environment on NT.</td>
<td>89</td>
</tr>
<tr>
<td>Setting up a source environment.</td>
<td>90</td>
</tr>
<tr>
<td>Installing the base NT software</td>
<td>91</td>
</tr>
<tr>
<td>Installing Microsoft Windows NT 4.0 server</td>
<td>91</td>
</tr>
<tr>
<td>Installing Microsoft Office 2000</td>
<td>92</td>
</tr>
<tr>
<td>Installing Adobe Acrobat</td>
<td>92</td>
</tr>
<tr>
<td>Installing Internet Explorer</td>
<td>92</td>
</tr>
<tr>
<td>Installing DB2 for NT</td>
<td>93</td>
</tr>
<tr>
<td>Creating the DB2 startup account</td>
<td>93</td>
</tr>
<tr>
<td>Installing the DB2 server</td>
<td>93</td>
</tr>
<tr>
<td>Installing the Siebel 2000 DB2 for NT Fix Pack</td>
<td>94</td>
</tr>
<tr>
<td>Verifying the DB2 installation</td>
<td>94</td>
</tr>
<tr>
<td>Configuring DB2 for Siebel 2000</td>
<td>95</td>
</tr>
<tr>
<td>DB2 Database Manager configuration parameters</td>
<td>96</td>
</tr>
<tr>
<td>Setting the DB2 environment parameters</td>
<td>97</td>
</tr>
<tr>
<td>Creating the initial DB2 database</td>
<td>97</td>
</tr>
<tr>
<td>Creating the skeleton Siebel database</td>
<td>98</td>
</tr>
<tr>
<td>Creating the initial buffer pools</td>
<td>98</td>
</tr>
<tr>
<td>Verifying the buffer pools</td>
<td>98</td>
</tr>
<tr>
<td>Creating the regular and temporary table spaces</td>
<td>99</td>
</tr>
<tr>
<td>Verifying the table spaces</td>
<td>99</td>
</tr>
<tr>
<td>Configuring the database</td>
<td>99</td>
</tr>
<tr>
<td>Creating required NT user accounts</td>
<td>101</td>
</tr>
<tr>
<td>Creating the Siebel File System</td>
<td>101</td>
</tr>
<tr>
<td>Installing the Siebel Gateway Server</td>
<td>102</td>
</tr>
<tr>
<td>Installing the Siebel Gateway Server</td>
<td>102</td>
</tr>
<tr>
<td>Starting the Siebel Gateway Server NT service</td>
<td>103</td>
</tr>
<tr>
<td>Inspecting the Gateway Server directory</td>
<td>103</td>
</tr>
<tr>
<td>Installing the Siebel Server</td>
<td>104</td>
</tr>
<tr>
<td>Installing the Siebel Server</td>
<td>104</td>
</tr>
<tr>
<td>Inspecting the Siebel Server directory</td>
<td>105</td>
</tr>
<tr>
<td>Starting the Siebel Server NT service</td>
<td>106</td>
</tr>
<tr>
<td>Verifying the Siebel Server ODBC data source</td>
<td>106</td>
</tr>
<tr>
<td>Installing the Siebel Database Server</td>
<td>107</td>
</tr>
<tr>
<td>Installing the Siebel Database Server scripts</td>
<td>107</td>
</tr>
<tr>
<td>Granting database privileges</td>
<td>107</td>
</tr>
<tr>
<td>Configuring the Siebel DB2 instance</td>
<td>108</td>
</tr>
<tr>
<td>Installing the stored procedures code</td>
<td>108</td>
</tr>
<tr>
<td>Initializing the Siebel database</td>
<td>108</td>
</tr>
<tr>
<td>Creating the Siebel schema</td>
<td>108</td>
</tr>
<tr>
<td>Verifying the results of install.ksh</td>
<td>111</td>
</tr>
</tbody>
</table>
Importing the Siebel object definitions ........................................ 111
Verifying the results of imprep.ksh ........................................... 112
Populating the Siebel File System ............................................ 112
Verifying the install ............................................................... 113
Installing the Siebel client ...................................................... 113
Testing the installation using the Siebel Call Center ...................... 114
Installing Siebel Tools ............................................................. 115
Detailed instructions ............................................................... 115
Testing Siebel Tools ............................................................... 116
Creating sample data .............................................................. 116

Appendix B. The .bat files ......................................................... 117
db2_db_mgr_cfg.bat .............................................................. 117
db2_db2set.bat ................................................................. 118
db2_dbcfg.bat ........................................................................ 118
tablespace.bat ................................................................. 119
forceit.bat ....................................................................... 120
bufferpool.bat ................................................................. 120

Appendix C. Creating sample data .............................................. 121
Entering data using Siebel Call Center ........................................ 121
Migrating the Siebel sample database ....................................... 124

Appendix D. DB2 customization jobs .......................................... 127
DSNZPARM parameters ........................................................... 128
Creating STOGROUPs .............................................................. 134
Creating the Siebel databases .................................................. 134
Creating the Siebel table spaces .............................................. 135
Creating Siebel tables and indexes .......................................... 153
Creating stored procedure objects ......................................... 158

Appendix E. DB2 DSNTIAUL and LOAD jobs .............................. 159
REXX EXEC to generate DSNTIAUL unload jobs ......................... 162
REXX EXEC to generate DB2 LOAD jobs .................................. 165

Appendix F. Validating object names .......................................... 169
Table names > 18 characters ...................................................... 169
Index names > 18 characters ..................................................... 169
Column names > 18 characters .................................................. 170
Index key > 255 bytes ............................................................ 170
The .dat file ........................................................................ 171

Appendix G. Dataexp/FTP/DB2 Load programs and jobs ............. 175
REXX procedure ................................................................. 175
Preface

This IBM Redbook will help you migrate the Siebel database from:

- DB2 for NT V6 to DB2 for OS/390 V6
- Oracle for NT V8.1.6 to DB2 for NT V6
- Oracle for NT V8.1.6 to DB2 for OS/390 V6

The book describes the migration experiences gained while migrating a Siebel database at the IBM ITSO Center in Poughkeepsie, New York, and at the IBM Hursley Laboratory in the UK.

It provides an overview of Siebel architecture, and introduces the migration methodology needed to move the Siebel database from a DB2/Oracle for NT platform to a DB2 for OS/390 platform. It provides a step-by-step description of the database migration process, and discusses in detail the different methods of moving data from one platform to the other.

This redbook will be especially useful for those migrating the Siebel database for the first time.

The team that wrote this redbook

This redbook was produced by a team of specialists from around the world working at the International Technical Support Organization Poughkeepsie Center.

Viviane Anavi-Chaput is a Senior IT Specialist for BI and CRM with the IBM International Technical Support Organization in Poughkeepsie, New York. She writes extensively, teaches worldwide, and presents at international conferences. Before joining the ITSO in 1999, Viviane was a Senior Data Management Consultant at IBM Europe, France. She was also an ITSO Specialist for DB2 at the San Jose Center during 1990-1994.

Miguel de Andrade is an Architecture Specialist for Siebel Systems in Egham, United Kingdom. He has 10 years of experience working mainly in DB2 for OS/390 as a DB2 DBA, DB2 Systems Programmer, and third-party DB2 tools specialist.

Henry Horst is a Certified Principal Consultant at Siebel Systems in the US. His areas of expertise include architectural designs and configurations, hardware/software integration, performance analysis and business planning.

Glenn Lacco is a Software Engineer for Business Innovation Services (BIS) in IBM, Sydney, Australia. He has 11 years of experience in application and database development. He holds a computing degree from University of Technology, Sydney. His areas of expertise include middleware development for systems integration and Oracle/UNIX application development.
Jeffrey Wiese is a Market Support Representative in IBM, US. He joined IBM in 1973 and was a PSR, instructor, SE, and Large Systems Marketing Specialist before joining the S/390 New Technology Center in Poughkeepsie, New York.

Marion Wilmot is a Senior Technical Support Engineer for Siebel Systems in the UK. She joined Siebel in 1998 with 20 years of previous experience in the IT industry; many years of this were spent in applications design and development using DB2 as the RDBMS. Her areas of expertise within Siebel include Siebel installation and upgrade.

Alex Yuen is an IT specialist in Sell and Support, Business and Innovation Services (BIS) in IBM, Sydney, Australia. He has 14 years of experience in the application development area. He holds a degree in Computer Science from University of Sydney. His areas of expertise include Data Warehouse and Customer Relationship Management.

Thanks to the following people for their contributions to this project:

Peter Bertolozzi
Richard Conway
Vassilis Karras
Mike Schwartz
IBM International Technical Support Organization, Poughkeepsie, New York

Sam Amsavelu
Palmer Brisendine
Sudha Dev
Louis Doran
Dennis Dutcavich
Kevin Fenning
Bonnie Lane
Paul Maxfield
IBM US

Robert Harbus
IBM DB2 UDB Data Management Consulting Services, Toronto, Canada

Eric Cicchiello
Richard Corrihons
Alain Poquillon
IBM Montpellier, France

Andy Anderson
Paul Fletcher
Simon Harris
Adrian Houselander
Thanks also to Al Schwab and Terry Barthel for their editorial assistance, and to Ella Buslovich for her graphics assistance.

**Special notice**

This publication is intended to help technical professionals who are migrating the Siebel database for the first time. The information in this publication is not intended as the specification of any programming interfaces that are provided by Siebel, OS/390 and DB2. See the PUBLICATIONS section of the IBM Programming Announcement for OS/390 and DB2, and Siebel documentation for more information about what publications are considered to be product documentation.
IBM Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States and/or other countries:

- e (logo)®
- IBM ®
- AIX
- AS/400
- AT
- CT
- Current
- DataJoiner
- DataPropagator
- DB2
- DB2 Connect
- DB2 Universal Database
- DFS
- DPI
- DRDA
- MVS
- OS/390
- PAL
- RACF
- RETAIN
- S/390
- Sequent
- SP
- XT
- z/OS
- 400

Comments welcome

Your comments are important to us!

We want our Redbooks to be as helpful as possible. Please send us your comments about this or other Redbooks in one of the following ways:

- Use the online **Contact us** review redbook form found at
  ibm.com/redbooks
- Send your comments in an Internet note to
  redbook@us.ibm.com
- Mail your comments to address on Page ii
Introducing Siebel database migration

This chapter gives an overview of Siebel architecture and configurations, and introduces our migration experience. It also describes the database architecture, which will help you to understand the database migration.

Although this environment may be well known to Siebel users and developers on NT or AIX platforms, the subject might be new to OS/390 system programmers and database administrators (DBAs), and requires some explanation in order to prepare the OS/390 audience to undertake a Siebel database migration to the OS/390 platform.

The chapter describes the following:

- Siebel architecture and configurations
- When should you migrate the Siebel database?
- Siebel database migration
- Siebel database structure
- Which tool should you use to move data?
- Methodology for migrating to DB2 for OS/390
- Migration scenarios
1.1 Siebel architecture and configurations

Siebel 2000 is an application solution for Customer Relationship Management (CRM). Figure 1-1 shows its logical architecture and physical n-tier configurations. The logical architecture has three layers: client, application, and data. Those three logical layers can be physically implemented on one platform (1-tier), or two platforms (2-tier), or three platforms (3-tier). The OS/390 can be used for the Siebel database in a three-tier configuration as shown in Figure 1-1.

For example, a customer environment might have developers working on 1-tier or 2-tier configurations, while test, integration and production are done on 2-tier configurations. If capacity needs increase because of, for example, a growing number of connected users or growing volumes of data, a customer may consider moving its production environment to a 3-tier implementation with the database on OS/390. Going from a 2-tier to a 3-tier configuration implies a database migration.

1.2 When should you migrate the Siebel database

Database migrations are undertaken because of a need for a higher level of performance, scalability, availability, integrity, and security required by corporate businesses. Integrating the capabilities of the OS/390, DB2, and Enterprise Storage Servers on the same platform can significantly improve Siebel performance, as explained in the following sections.
Need for performance and scalability
Capacity needs due to growth in either the number of users or in the amount of data could drive a customer to consider moving to DB2 for OS/390, which is acknowledged to be a highly scalable platform, capable of supporting in excess of 50,000 users. DB2 supports very large databases. It allows the parallel processing needed to process large volumes of data—especially to load, back up, reorganize, and recover the data.

Need for reliability, availability, and serviceability
The OS/390 computing platform is unmatched in its proven ability to support mission-critical business applications. OS/390 has many system management capabilities, providing a very high level of data security, integrity, and the ability to manage diverse workloads.

Need for server consolidations
One of the key selling points of the Siebel application is that it gives your employees a consolidated enterprise view of customer databases; any Call Center rep or sales rep can see customer information. This is not possible if you have to split your database because you need multiple servers.

Need to move off a competitive platform
A customer may want to change hardware vendors (moving off a competitive platform to an IBM platform) which could also drive a change of the RDBMS. The cost of deploying a large database supporting very high numbers of users on a single DB2 for OS/390 database instance can be significantly less than the cost of implementing and managing multiple database instances on multiple machines, especially when leveraging an existing hardware and support infrastructure.

1.3 Siebel database migration
Figure 1-2 on page 4 shows a typical Siebel application implementation including five Siebel Enterprise environments: development, test, integration, production, and new production.

In this example, each Siebel Enterprise environment is a 2-tier implementation except for the new production environment, which is a 3-tier implementation with the Siebel database moving to the DB2 for OS/390 platform. This redbook focuses on the database migration.
Figure 1-2  Siebel implementation with database migration

Figure 1-2 also positions the Siebel databases and repositories used in this implementation. Notice that the repository resides in the database (148 tables) and also has a compiled version residing on a flat file in the application server and the connected clients. The repository is further explained in 1.4.1, “The repository” on page 4.

### 1.4 Siebel database structure

Before you undertake a database migration to OS/390, it helps to understand the Siebel database structure; we describe it in the following sections.

The Siebel application uses relational databases to store data. Nevertheless, the data model is controlled directly by the Siebel application (and not by the RDBMS) using a repository. By controlling the data model, the Siebel application relieves itself of RDBMS product dependency, so the Siebel code gains the possibility of porting on different RDBMSs.

#### 1.4.1 The repository

Siebel Systems uses a repository to control its data model. The repository contains, among other things, the description of the database objects, such as tables, indexes, and control information, to manage referential integrity and dependency between data. To access data in the database, the Siebel application must first check the data model control information in the repository.
The repository resides in the database (148 tables). To improve access performance to the repository, there is also a binary representation of the repository, called the .srf file, which is compiled and stored on each Siebel Enterprise environment.

When newly developed applications trigger schema changes, those modifications must be reflected in the production repository as well as in the physical database schema before the application goes on production. Siebel Tools are used to customize the repository. This is done by checking out a copy of the repository from the application server to a Siebel Tools environment residing on a client workstation. Here, customizations can be made (for example, presentation enhancements like applet component additions, or data schema changes such as adding columns to a table). These changes to the repository are propagated to the application server site, where they are integrated with other development activity when the repository is checked back in. The dev2prod utility is used to migrate the customized repository to production, as well as to identify and record the schema changes.

### 1.4.2 The data

When migrating the Siebel database from one RDBMS to another (such as Oracle to DB2), or moving the same RDBMS to a different hardware platform (such as Oracle/DB2 for NT to DB2 for OS/390), the issue of moving data presents itself. During a database migration, three categories of data need to be addressed:

- Repository data
- Seed data
- User data

### Migrating repository data

Migrating repository data does not present any challenges because dev2prod is a commonly used procedure in the development environment. Customizing the Siebel database for new applications makes moving the repository a regular occurrence, since it has to be updated to enable these changes.

**Note:** Keep in mind that the Siebel .srf file resides on a file system, not in the RDBMS, and the source and target repositories should be kept in sync.
Migrating seed data
Seed data doesn’t get moved unless you migrate the database. Seed data refers to data that maps “users to responsibilities”, version information, and other control data used by the Siebel database to manage itself. Basic seed data is moved into specific seed data tables, named S_xxx etc, during the Siebel installation process.

Seed data also evolves and changes over time. Therefore, when you are migrating the database, you must also migrate the seed data to the new target RDBMS.

Migrating user data
Moving user data presents the biggest challenge due to its large volumes.

1.5 Which tool should you use to move data

There are several tools you can use to move data: there are Siebel tools, and there are the RDBMS tools on which the Siebel database resides. The choice of which technique to use is based on the kind of migration to be performed and the volumes of data involved. We can categorize those tools as follows.

1.5.1 Siebel database utilities
Using Siebel database utilities offers the advantage that they are well known by Siebel Systems and if you have any issues during your migration, you can request assistance from Siebel.

Dataexp/Dataimp utilities
These export/import utilities are currently reserved for Siebel Technical Services usage, and obtaining access to them might require involvement by Siebel. Those utilities are not currently documented in Siebel manuals.

Dataexp/Dataimp work very well in every situation except perhaps for very large volumes of data. Nevertheless, you must be careful with the referential integrity issue. You should copy all tables to make sure you will not corrupt any foreign keys and their dependencies.

Siebel Enterprise Integration Manager
Siebel Enterprise Integration Manager (EIM) is fully supported by Siebel Systems and guarantees the referential integrity of the Siebel data, but it can be complex to implement.
The complexity derives from the fact that some base tables get mapped to several interface tables, so you need to determine which interface tables are appropriate for the table you want to migrate. You need to prepare your migration carefully and run tests on small volumes of data to make sure your data will be migrated correctly.

EIM is suitable for small or large volumes of data.

1.5.2 RDBMS-specific database utilities

These utilities are fast and offer the advantage of greater efficiency in moving large data volumes. The tables you are moving may need the same column definitions at both ends. This procedure should be thoroughly tested beforehand to make sure there will be no referential integrity issues.

1.5.3 Data warehousing tools

There are many data warehousing tools on the market, such as IBM DB2 Warehouse Manager, IBM DataPropagator, ETI, and Sagent, to name a few. For a one-time migration, those tools might be difficult to justify unless you already have them in-house, in which case they can be very helpful.

1.6 Methodology for migrating to DB2 for OS/390

The recommended method for migrating the Siebel database to DB2 for OS/390 is as follows:

1. First, install a standard, out-of-the-box, new target production environment including a Siebel Server on NT and a Siebel-DB2 database on OS/390 and make sure the environment works as expected. (Refer to the IBM Redbook *Siebel 2000 Database Implementation on OS/390 Using NT Siebel Servers*, SG24-5953, which provides a detailed, step-by-step description of this installation).

2. Next, migrate the data from the source Oracle/DB2 for NT environment to the target DB2 for OS/390 environment.

   Note that it is strongly recommended that you migrate from one equivalent version/release to another. For example, if you want your target system to be at Version 6 level, but your source system is backlevel, you should first upgrade your source system to Version 6 level before you start the migration process. You should not attempt to migrate and upgrade at the same time.

This method allows a phased migration, which diminishes the risk of errors and simplifies problem determination.
1.7 Migration scenarios

In this redbook, we consider migrations to the DB2 for OS/390 from two different source environments:

- DB2 for NT to DB2 for OS/390
- Oracle for NT to DB2 for OS/390.

We describe step-by-step the following scenarios:

- We first migrated DB2 for NT to DB2 for OS/390. This is a one-phase direct migration.
- We next migrated Oracle for NT to DB2 for OS/390 in two ways:
  - Two-phase approach: Oracle for NT to DB2 for NT to DB2 for OS/390
  - One-phase approach: Oracle for NT to DB2 for OS/390 directly

We first needed to set up a source NT test environment from which to migrate. For reference purposes, we document the implementation of the Siebel source environment on NT, using a DB2 for NT database, in Appendix A, “Implementing the Siebel source environment on NT” on page 89. Those who already have a Siebel source environment to test with will obviously not need this section. On the other hand, the section can be very useful for those who want to create a test source environment for database migration purposes.

The next step was to set up the target migration environment and migrate the data. We did the following:

- We set up a new out-of-the-box target environment using a DB2 for OS/390 database as described in Chapter 2, “Implementing the target Siebel environment on OS/390” on page 11. This chapter covers:
  - NT Siebel Server configuration to support the Siebel database on OS/390
  - Siebel database configuration on DB2 for OS/390.

This chapter is actually an abstract of the IBM Redbook *Siebel 2000 Database Implementation on OS/390 Using NT Siebel Servers*, SG24-5953, which details this installation.

- We migrate the data from the DB2 for NT source environment to the target DB2 for OS/390 environment as described in Chapter 3, “Migrating Siebel data from DB2 for NT to DB2 for OS/390” on page 31. This chapter covers the following migration techniques:
  - Siebel Dataexp/Dataimp
  - Siebel EIM
  - IBM DB2 tools

- Next, we look into Oracle for NT migration to DB2 for OS/390. We start with the two-phase migration.
The first migration phase is to move the data from Oracle for NT to DB2 for NT as described in Chapter 4, “Migrating Siebel data from Oracle for NT to DB2 for NT” on page 53. This chapter covers the following migration techniques:

- Siebel Dataexp/Dataimp with and without schema changes
- Siebel EIM

The second migration phase is to move the data from DB2 for NT to DB2 for OS/390, which is described in Chapter 3, “Migrating Siebel data from DB2 for NT to DB2 for OS/390” on page 31.

We then look into one-phase migrations of Oracle for NT to DB2 for OS/390 as described in Chapter 5, “Migrating Siebel data from Oracle for NT to DB2 for OS/390” on page 73. This chapter covers the following migration techniques:

- Dataexp/Dataimp with and without schema changes
- Dataexp/ftp/DB2 Load
- DB2 Relational Connect/DB2 Load
Migrating Siebel Database from DB2/Oracle for NT to DB2 for OS/390
Implementing the target Siebel environment on OS/390

This chapter describes the implementation of the target Siebel environment to which we are going to migrate.

It covers the following sections:

- Preparing the OS/390 environment for the Siebel database
  OS/390, RACF, and DB2 customizations
- Establishing database connectivity to DB2 for OS/390
  - DDF setup
  - DB2 Connect EE setup
- Installing Siebel 2000 on NT
- Installing the Siebel-DB2 database on OS/390
- Installing the Siebel connected client
- Testing the installation using the Siebel Call Center
We describe the procedures we used to install the Siebel Server on NT, supporting the Siebel database on DB2 for OS/390. See Figure 2-1.

This is a standard, out-of-the-box, new target environment with a Siebel database on OS/390. In this chapter, we give a high-level abstract of the implementation procedures, with the specifics for Siebel V6.2.1. Detailed explanations of how to implement such an environment can be found in the redbook *Siebel 2000 Database Implementation on OS/390 Using NT Siebel Servers*, SG24-5953.
2.1 Preparing the OS/390 environment for the Siebel database

You need to prepare and customize the OS/390 environment before you start implementing Siebel 2000. The following documentation will help you plan for this installation:

- *Siebel 2000 Database Implementation on OS/390 Using NT Siebel Servers*, SG24-5953

You also need the help of the OS/390 database administrator (DBA) to manually perform certain tasks described in this section on the OS/390 platform.

We assume DB2 V6 is installed on OS/390. Customizing DB2 and OS/390 environments includes the following.

2.1.1 Checking the maintenance level

We recommend that you provide a separate DB2 subsystem on OS/390 for the Siebel database for ease of maintenance.

Ensure that all maintenance is applied to the DB2 V6 subsystem. Our DB2 subsystem was at PUT level 0012. Make sure the required PTFs have been applied to your system, otherwise you may not enable the Siebel database. Check especially for:

- PTF UQ38405 with APAR PQ30652 (includes identity columns support)
- APAR II12440 (includes several Siebel requirements for DB2 for OS/390)

2.1.2 Creating RACF user IDs

You need to create RACF user IDs and grant them DB2 authorization. These user IDs are used by the Siebel application to connect to the DB2 for OS/390 Siebel database. You need to perform the following tasks:

- Enable DB2 secondary authorization IDs.
- Create RACF group IDs SSEROLE and SSEEIM.
  - The SSEROLE group has Siebel user privileges. SSEROLE has DB2 authority (select, insert, update, and delete authority on all tables) granted to it when the Siebel install scripts are run.
  - The SSEEIM group has EIM user privileges (use of storage groups and buffer pools for EIM temporary tables).
- Create RACF user IDs SIEBEL and SADMIN.
  - SIEBEL has table owner privileges.
– SADMIN has DB2 administrator privileges (DBADM).
– Connect both SIEBEL and SADMIN user IDs to RACF group SSEROLE.

Note that DB2 authority will be granted automatically to the group ID SSEROLE during the installation of Siebel database scripts. The user ID SIEBEL will be used as the table owner, and the TSO session will be started using the SIEBEL user ID. The TSO session will be used to submit the DDL to create the SIEBEL database objects.

2.1.3 Creating a large VTOC
When many indexes are created (which are small VSAM files), it is possible that you will fill up the VTOC before filling up the DASD space available. Review your VTOC space after you have completed your installation.

2.1.4 Creating DB2 storage groups
The DB2 DBA must create the STOGROUPs and grant use of the storage groups to the SIEBEL table owner ID and the SSEEIM group. There is no Siebel script to help you build and run this job. Refer to “Creating STOGROUPs” on page 134 to see the job we built.

For higher performance, consider using multiple storage groups, particularly for the separation of indexes and data.

2.1.5 Customizing DB2 buffer pools
Make sure that buffer pools used by the Siebel database are active. The Siebel database uses 4K, 16K and 32K tablespaces, so the DBA needs to make sure that these buffer pools have been activated. Buffer pools are activated by altering the buffer pool and assigning a VPSIZE>0.

Grant use of these buffer pools to the SIEBEL table owner and to the SSEEIM group.

For higher performance, consider using a separate buffer pool for the indexes.

2.1.6 Customizing the EDM pool
Ensure that the EDM pool size is large enough to handle the Siebel DBDs.

2.1.7 Customizing DSNZPARM parameters
Customize the following DSNZPARM and DSNHDECP parameters as required by Siebel:
**ENSCHEME (ASCII)**

Is the system wide option and implies that all applications on this system use ASCII data. If you want more flexibility and allow other applications to use EBCDIC data, you can leave ENSCHEME (EBCDIC) in the DSNZPARM and specify CCSID (ASCII) on the CREATE DATABASE command for all Siebel databases.

**CACHEDYN (YES)**

Use of dynamic SQL caching improves performance.

**SPRMMXT (40)**

Maximum number of tables in a join.

**DSMAX (21000)**

Maximum number of open data sets.

**IDLE**

Idle thread time-out should be used.

**CMTSTAT (INACTIVE)**

Activates thread/connection pooling with DB2. If your installation must support a large number of connections, specify INACTIVE.

**MAXDBAT (500)**

Maximum number of database access threads (DBATs) that can be active concurrently. 500 is recommended for a large number of users (19K users, for example).

**CONDBAT (150000)**

Maximum number of concurrent remote connections. When the maximum is reached, the connection request is rejected. You can go up to 150000.

Refer to “DSNZPARM parameters” on page 128 to see the values we used for DSNZPARM.

### 2.1.8 Enabling DB2 stored procedures

Siebel EIM processes use DB2 stored procedures to trigger the DB2 RUNSTATS utility, when required in the middle of an EIM run, to update the statistics of the EIM tables.

To enable DB2 stored procedures, verify the following:

- The temporary database with three table spaces (4K, 16K and 32K) required to enable DB2 stored procedures has been created.
- DSNUTILS is installed and operational.
- The OS/390 Workload Manager (WLM) is enabled and running in GOAL mode, as required, to support DB2 stored procedures.
- OS/390 Resource Recovery Services (RRS) are enabled to support DB2 stored procedures.
Refer to “Creating stored procedure objects” on page 158 to see the jobs we built to create the stored procedure objects.

For more information on WLM and RRS setup, refer to redbook Siebel 2000 Database Implementation on OS/390 Using NT Siebel Servers, SG24-5953.

2.2 Enabling database connections to DB2 for OS/390

The following tasks are required to enable these connections.

2.2.1 DDF setup on OS/390

The DB2 subsystem on OS/390 uses the location name DB2D. This DB2D instance is our target database for the Siebel database.

In our system, DDF listens on TCP/IP port 33322. DDF must be up and running for the connection to be established.

2.2.2 DB2 Connect EE setup on NT

On the NT server you do not need to install a full DB2 EE. You only need to install DB2 Connect EE to establish connectivity between the Siebel Server on NT and the Siebel database on OS/390. Refer to Siebel 2000 Database Implementation on OS/390 Using NT Siebel Servers, SG24-5953, for information about installing DB2 Connect EE.

During this process we created the administrator ID sadmin/sadmin.

After successfully installing DB2 Connect EE (including the Fix Pack), you now need to configure it.

To establish an ODBC connection between DB2 Connect EE and DB2 for OS/390, go to Start -> Programs -> DB2 for Windows NT -> Client Configuration Assistant to create and catalog database descriptors. On the welcome screen, choose Add Database and select the following options:

1. For the Source, choose Manually configure a connection to a DB2 database.
2. For the Protocol, choose TCP/IP, and OS/390 as the target operating system.
3. For the TCP/IP configuration, enter the OS/390 fully qualified hostname or the IP address of the OS/390 host. Enter the port that is reserved for the DDF in the DB2 instance. In our case, the host name is wtsc04.itso.ibm.com and the port is 33322.
4. For the Database, add the Location name, in our case DB2D, which is the DB2 OS/390 instance name where we want to connect. Enter the Database alias name that will be used by DB2 Run-Time clients to connect to the DB2 OS/390 instance. We choose DB2DSB.

5. For ODBC, select Register this database for ODBC source, and As a system data source. This will allow DB2 Run-Time clients to access the host database.

6. For Node Options, click Next.

7. For security Options, select Configure security options and choose On the host or AS/400 (DCS).

8. Skip Host or AS/400 Options, and click Done.

When you get the confirmation message that the connection configuration for DB2DSB is added successfully, you are ready to test the connection. To verify that DB2D and its DDF are up and running, click Test Connection.

You are prompted to enter a valid TSO user ID and password. The user ID should also be DB2-authorized. We used the user ID/password SADMIN/SADMIN.

You should get a message saying that our connection is successful. If you do not receive this message, you have to check that you can ping the host and that DDF is running.

2.2.3 Client setup on a PC

The next step is to set up a client PC. We choose to install DB2 Run-Time client to connect the PC to the NT where we installed the DB2 Connect EE. Next we tested the client connection to DB2 for OS/390 by issuing the following commands:

```
db2start
connect to db2dsb user sadmin using sadmin
```

A successful connection results in the following message:

```
Database Connection Information
    Database server = DB2 OS/390 6.1
    SQL Authorization ID = sadmin
    Local Database Alias = db2dsb
```
2.3 Installing Siebel 2000 on the NT server

For this installation, either follow the instructions in Siebel 2000 Database Implementation on OS/390 Using NT Siebel Servers, SG24-5953, or refer to Appendix A, “Implementing the Siebel source environment on NT” on page 89.

2.3.1 Creating required NT user accounts

Create the administrator ID SIEBEL.

Add SIEBEL and SADMIN to RACF group SSEROLE.

2.3.2 Creating the Siebel File System

The file system should be sharable in read and write mode.

2.3.3 Installing the Siebel Gateway Server

We used the NT Administrator account value of ITSONT0\sadmin (domain\username).

2.3.4 Installing the Siebel Server

For this installation:

- Our Gateway server address is ITSONT0.
- The enterprise server is Siebel_390.
- Our Database Alias is DB2DSB.
- The table owner is SIEBEL.
- Database username/password is sadmin/sadmin.
- Siebel Server name is ITSONT0.
- The Synch Manager port has the default value 40400.

2.4 Installing the Siebel Database Server on OS/390

Installing the database server on OS/390 involves the following tasks.

2.4.1 Installing the database server scripts

Insert the Siebel Server CD-ROM, navigate to the dbsrvr directory, and double-click setup.exe to start the installation of the database server scripts.
2.4.2 Creating the Siebel physical schema on OS/390

At this point, you must have done your preparations on OS/390, as follows:

- Your RACF user IDs and groups are defined and authorized appropriately.
  You will use those user IDs to connect from the NT server to the DB2 for OS/390 using an ODBC/DRDA connection.
- The DB2 for OS/390 settings are customized.
- DB2 storage groups are defined.
- You have an ODBC connection established between the NT server and the DB2 for OS/390, using the connection name DB2DSB.

The next step is to create the Siebel-DB2 databases, table spaces, tables and indexes on OS/390.

Creating the Siebel-DB2 databases

The DB2 DBA must create ten Siebel-DB2 databases. The Siebel application does not provide a script to help you build and execute this job. Refer to “Creating the Siebel databases” on page 134 to see the job we built.

To figure out the names of the ten databases, we looked in the sample configuration files provided by Siebel Systems to build the table spaces (tbspaces.sql and tbspaces.ctl).

The database prefixes/names used for the table spaces in the samples were XXXXX001 through XXXXX010. We chose to replace the Xs with a more significant name and used the database names SIEBEL01 to SIEBEL10 in the job we ran to create them.

Creating the Siebel-DB2 table spaces

Siebel Systems provides installation scripts that generate the DDL to create the table spaces and tables required for the Siebel application. The layout in V6.2.1 is different from the previous versions/releases.

The file sea621\dbsrvr\db2390\tbspaces.sql must be changed to specify the correct database name, buffer pool name, storage group, and parameters, then uploaded (using FTP, for example) to the OS/390 TSO session and submitted. Refer to “Creating the Siebel table spaces” on page 135 to see the job we submitted to create the Siebel-DB2 table spaces.

A DB2 for OS/390 DBA must review the table space parameters to adjust them according to the particular shop requirements.
Segmented table spaces

Defining table spaces as segmented can speed up both inserts and deletes. Currently, Siebel Systems recommends using segmented table spaces, as shown in the following example:

```sql
CREATE TABLESPACE FFFFD001
  IN SIEBEL1
  SEGSIZE 32 USING STOGROUP STONGNP01
  BUFFERPOOL BP32K1
  LOCKSIZE PAGE;

COMMIT;
```

Segsize

Several factors need to be considered in determining the right segsize:

- The size of the repository tables depends on the Industry Vertical applications selected and user-specific customizations.
- Other tables contain user data or are empty. Empty tables differ according to the Industry Vertical application installed. The size of user data depends on the user’s specific use of the Siebel application.
- A user can also move tables around, which complicates things, because those tables may end up in different table spaces than they were originally assigned to. Consequently, for instance, if the largest table happens to be moved out, the remaining tables of the table space will incur a completely different size attribute.

It is, therefore, not easy to recommend a segsize. Much depends on user customizations, and DBAs usually have their own opinion on what value to choose.

In our test environment, we chose a segsize of 32. We wanted a segsize that would make it possible to fetch the maximum number of pages per prefetch, and that would be the most efficient for the majority of the tables. For small tables that may only contain a few rows, having a large segsize will use more DASD than required. On the other hand, a large segsize might waste some disk space but maintain a better physical clustering on large tables. The trade-off is between choosing small tables that efficiently use disk space, or maintaining a higher level of clustering for larger tables which will have a bigger impact on performance.
DB2 Administration Guide has the following recommendations on segsizes.

<table>
<thead>
<tr>
<th>Number of pages</th>
<th>SEGSIZE recommendation</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;= 28</td>
<td>4 to 28</td>
</tr>
<tr>
<td>&gt; 28 &lt;128</td>
<td>32</td>
</tr>
<tr>
<td>&gt;= 128</td>
<td>64</td>
</tr>
</tbody>
</table>

We made the assumption that most Siebel tables would be in the <128 page range, and thought a segsize of 32 would be a safe starting point. Production environments should nevertheless evaluate their table sizes and determine if a segsize of 64 might be more appropriate for some tables.

PRIQTY/SECQTY

PRIQTY and SECQTY quantities must be specified when the tablespaces and indexes are created. The install will not work if the PRIQTY and SECQTY values are defaulted.

Coming up with the exact PRIQTY/SECQTY values can become very involved depending on how precise you want to be. The following is a relatively simple approach to come up with a rough estimate for PRIQTY and SECQTY. You may want to put the following information in a spreadsheet or DB2 table.

- Get a row count of each table that will be migrated.
- Estimate the number of rows that will fit on a page, something like 15 rows per page, and divide the rows by the number of rows per page to determine the number of 4K pages that will be required. Picking a number like 15 should allow for free space for most tables.
- Multiply the number of pages you come up with by 4. The 4 represents a 4K page and you want to determine the number of 1K blocks that are required.
  
  Now you can get more precise if you also identify whether a table is in a 4K, 16K, or 32K tablespace, then multiply the number by 1, 4, or 8, respectively, but for this quick estimate that may not be necessary.

  The number you come up with can be used as the PRIQTY value. Remember that the PRIQTY value in the CREATE TABLESPACE DDL statement is in 1K blocks.

- Choose 25% - 50% of the PRIQTY value as the SECQTY value, depending on the initial size of PRIQTY.

- Using the DDL that gets generated by the Siebel system, record the tablespaces each table is assigned to. Calculate the sum of the PRIQTY values for all the tables in the same tablespaces. Do the same for the
SECQTY values. Use the calculated PRIQYT and SECQTY values in the CREATE TABLESPACE statements.

The important thing is to cover the largest tables. Normally, these will be in their own tablespaces; if not, they should be placed in their own tablespaces.

The PRIQTY/SECQTY values are specified in 1024 1K blocks, which has already been taken care of by multiplying by 4 rather than 4096.

There are more scientific ways of calculating the PRIQTY and SECQTY values, which are documented in the DB2 manuals based on actual row size, but since Siebel uses VARCHARS that are not filled to capacity, the more scientific approach is more complex and may not give you any better results.

- Take a percentage of the PRIQTY/SECQTY values calculated for a table, for example 20%-40%, depending on the size of PRIQTY, and use this value for the PRIQTY values in the creation of the table’s indexes. Use 25%-50% of the indexes’ PRIQTY as the SECQTY value.

- The minimum size for tablespaces and indexes should be 720 for PRIQTY/SECQTY.

For exporting the repository database SIEBEL05, Siebel recommends the following PRIQTY and SECQTY values in the CREATE statement:

```sql
CREATE TABLESPACE HHHHH001
    IN SIEBEL05
    SEGSIZE 32 USING STOGROUP STOGRP01
    PRIQTY 7200 SECQTY 720
    BUFFERPOOL BP16K1
    LOCKSIZE PAGE;
COMMIT;
```

**Buffer pools**

If you are planning to migrate customized tables, with Long Varchar columns for example, data might not fit into the 4K or 16K table spaces defined at the target OS/390 database. You will have to alter those table spaces to allocate 16K or 32K buffer pools to them.

**Commit**

We also recommend that you issue a COMMIT after each CREATE statement. This allows a restart from the last commit point in case of failure of the CREATE job.
**Database, table space and buffer pool layout**

Table 2-1 shows the database, table space, and buffer pool layout for Siebel V6.2.1 in our installation, which includes 10 databases and 144 table spaces.

<table>
<thead>
<tr>
<th>Databases</th>
<th>Table spaces</th>
<th>Buffer pools</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIEBEL01</td>
<td>FFFFF001</td>
<td>BP32K1</td>
</tr>
<tr>
<td>SIEBEL02</td>
<td>EEEEE001</td>
<td>BP16K1</td>
</tr>
<tr>
<td>SIEBEL03</td>
<td>GGGGG010 to</td>
<td>BP16K1</td>
</tr>
<tr>
<td></td>
<td>GGGGG017</td>
<td></td>
</tr>
<tr>
<td>SIEBEL04</td>
<td>GGGGG001 to</td>
<td>BP16K1</td>
</tr>
<tr>
<td></td>
<td>GGGGG008</td>
<td></td>
</tr>
<tr>
<td>SIEBEL05</td>
<td>HHHHH001 to</td>
<td>BP16K1</td>
</tr>
<tr>
<td></td>
<td>HHHHH002</td>
<td></td>
</tr>
<tr>
<td>SIEBEL06</td>
<td>BBBB024 to</td>
<td>BP1</td>
</tr>
<tr>
<td></td>
<td>BBBB029</td>
<td>BP1</td>
</tr>
<tr>
<td></td>
<td>BBBB031 to</td>
<td>BP16K1</td>
</tr>
<tr>
<td></td>
<td>BBBB044</td>
<td>BP1</td>
</tr>
<tr>
<td></td>
<td>CCCCC002</td>
<td>BP1</td>
</tr>
<tr>
<td></td>
<td>DDDDD015 to</td>
<td>BP1</td>
</tr>
<tr>
<td></td>
<td>DDDDD024</td>
<td>BP1</td>
</tr>
<tr>
<td></td>
<td>DDDDD055</td>
<td>BP1</td>
</tr>
<tr>
<td>SIEBEL07</td>
<td>AAAAA001 to</td>
<td>BP1</td>
</tr>
<tr>
<td></td>
<td>AAAAA005</td>
<td>BP1</td>
</tr>
<tr>
<td></td>
<td>BBBB045 to</td>
<td>BP1</td>
</tr>
<tr>
<td></td>
<td>BBBB060</td>
<td>BP1</td>
</tr>
<tr>
<td></td>
<td>DDDDD005</td>
<td>BP1</td>
</tr>
<tr>
<td></td>
<td>DDDDD014</td>
<td>BP1</td>
</tr>
<tr>
<td>SIEBEL08</td>
<td>BBBB001 to</td>
<td>BP1</td>
</tr>
<tr>
<td></td>
<td>BBBB004</td>
<td>BP1</td>
</tr>
<tr>
<td></td>
<td>BBBB061 to</td>
<td>BP16K1</td>
</tr>
<tr>
<td></td>
<td>BBBB062</td>
<td>BP1</td>
</tr>
<tr>
<td></td>
<td>CCCCC001</td>
<td>BP1</td>
</tr>
<tr>
<td></td>
<td>DDDDD001 to</td>
<td>BP1</td>
</tr>
<tr>
<td></td>
<td>DDDDD003</td>
<td>BP1</td>
</tr>
<tr>
<td></td>
<td>DDDDD041 to</td>
<td>BP1</td>
</tr>
<tr>
<td></td>
<td>DDDDD054</td>
<td>BP1</td>
</tr>
<tr>
<td>SIEBEL09</td>
<td>BBBB006 to</td>
<td>BP1</td>
</tr>
<tr>
<td></td>
<td>BBBB016</td>
<td>BP1</td>
</tr>
<tr>
<td></td>
<td>DDDDD030 to</td>
<td>BP1</td>
</tr>
<tr>
<td></td>
<td>DDDDD040</td>
<td>BP1</td>
</tr>
</tbody>
</table>
Create Siebel tables and indexes on OS/390

Creating Siebel tables and indexes on the DB2 for OS/390 platform requires the generation of a DDL script. This script will create the tables and indexes, and grant the appropriate authorities.

Navigate to the d:\sea621\dbsrvr\db2390 directory and edit the generate_ddl.ksh script to contain the parameter values shown in Table 2-2. Refer to “Editing the Database Schema Generation Script” in Chapter 12 of Siebel Server Installation Guide, Version 6.2.

Table 2-2   The generate_ddl.ksh parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SRC_TBLO</td>
<td>SIEBEL</td>
</tr>
<tr>
<td>SRC_TBLO_PSWD</td>
<td>SIEBEL</td>
</tr>
<tr>
<td>ODBC</td>
<td>SiebServer_Siebel_390</td>
</tr>
<tr>
<td>DBSRVR_ROOT</td>
<td>d:\sea621\dbsrvr</td>
</tr>
<tr>
<td>SIEBEL_HOME</td>
<td>d:\sea621\dbsrvr</td>
</tr>
<tr>
<td>MODE</td>
<td>Install</td>
</tr>
<tr>
<td>MERGE_FLG</td>
<td>N</td>
</tr>
<tr>
<td>DO_DDL</td>
<td>N</td>
</tr>
<tr>
<td>TBSPACE_FILE</td>
<td>tbspaces.ctl</td>
</tr>
</tbody>
</table>

Execute the generate_dll script. When the script has run, the parameter values are displayed for confirmation. Reedit the file, if necessary, or enter Y to continue.

The generate_dll.ksh script generates a schema file named sieb_schema.sql.

This file contains the DDL to create tables and indexes and grant the appropriate database authorities. 1372 tables and 7327 indexes are created.
Figure 2-2 shows the information given by the script when successful execution took place.

```
TABLES CREATED : 1372
TABLES GRANTED : 1372
TABLES DELETED  : 0
TABLES MERGED   : 0
TABLES REBUILT  : 0
TABLES IGNORED : 0
COLUMNS ADDED   : 0
COLUMNS DELETED : 0
COLUMNS MODIFIED: 0
INDEXES CREATED : 7327
INDEXES DELETED : 0
INDEXES MODIFIED: 0
TOTAL ERRORS    : 0
Disconnecting from the database.
```

Review the file ddlimp.log for any errors. Be very careful with index truncation errors. DB2 for OS/390 limits the index length to 250 characters, which is shorter than the index length limit in DB2 for AIX/NT and Oracle. Ddlimp.log shows the truncated indexes. _U1 index truncation may affect performance. The error.rtf file contains a list of acceptable errors, including truncated indexes, and should be consulted when reviewing the Siebel log files.

Ship the sieb_schema.sql file to the Siebel DB2 for OS/390 host. If using FTP, the command line will appear as in Figure 2-3.

```
D:\sea621\dsrvr\DB2390>ftp wtsc04.itso.ibm.com
Connected to wtsc04.itso.ibm.com.
220 FPDMSU1 IBM FTP CS U2RT10 at wtsc04oa.itso.ibm.com, 19:52:44
220 Connection will close if idle for more than 5 minutes.
User (wtsc04.itso.ibm.com:(none)): siebel
331 Send password please.
Password:
230 SIEBEL is logged on. Working directory is "SIEBEL."
ftp> put sieb_schema.sql 'siebel.ftp.out3'
200 Port request OK.
125 Storing data set SIEBEL.FTP.OUT3
250 Transfer completed successfully.
4192324 bytes sent in 4.94 seconds (849.16 Kbytes/sec)
ftp>
```

Chapter 2. Implementing the target Siebel environment on OS/390  25
Collaborate with the DB2 for OS/390 DBA to execute this script on the OS/390 host to create the Siebel database schema. Refer to “Creating Siebel tables and indexes” on page 153 to see the DDL we executed on OS/390.

### 2.4.3 Importing seed data

Siebel Systems provides the install_siebel.ksh utility to load initial business data and establish the initial database views. Navigate to the d:\sea621\dbsrvr\DB2390 directory and edit the install_siebel.ksh script, i.e., update the parameter values according to Table 2-3. This script establishes initial Siebel table data and views.

Table 2-3  Install_siebel.ksh script parameter values

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SRC_USR</td>
<td>SADMIN</td>
</tr>
<tr>
<td>SRC_PSWD</td>
<td>SADMIN</td>
</tr>
<tr>
<td>SRC_TBLO</td>
<td>SIEBEL</td>
</tr>
<tr>
<td>SRC_TBLO_PSWD</td>
<td>SIEBEL</td>
</tr>
<tr>
<td>ODBC</td>
<td>SiebServer_Siebel_390</td>
</tr>
<tr>
<td>DBSRVR_ROOT</td>
<td>d:\sea621\dbsrvr</td>
</tr>
<tr>
<td>DO_DDL</td>
<td>Y</td>
</tr>
<tr>
<td>DB_LANG</td>
<td>enu</td>
</tr>
<tr>
<td>SIEBEL_HOME</td>
<td>d:\sea621\siebsrvr</td>
</tr>
</tbody>
</table>

Invoke the UNIX shell by entering `sh` at the DOS prompt.

Execute the script by typing `./install_siebel.ksh` at the UNIX shell prompt.

The script will prompt you to confirm the parameter values. Enter `Y` to confirm and submit the script for execution.

**Validating the database objects**

Siebel Systems supplies the validate_objects.ksh script to validate the views that have been created when the install_siebel.ksh script executed.
Edit the validate_objects.ksh script to contain the parameter values shown in Table 2-4.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SRC_TBLO</td>
<td>SIEBEL</td>
</tr>
<tr>
<td>SRC_TBLO_PSWD</td>
<td>SIEBEL</td>
</tr>
<tr>
<td>ODBC</td>
<td>SiebServer_Siebel_390</td>
</tr>
<tr>
<td>SEIBEL_HOME</td>
<td>d:/sea621/siebsrvr</td>
</tr>
</tbody>
</table>

Invoke the UNIX shell by typing `sh` at the DOS prompt.

Enter `.execute` to run the validate_objects.ksh script to validate schema. The script will prompt for confirmation of the parameter values. When executed, this script generates the report 390_objects.txt. Review this report to make sure that the following database views exist:

- S_PERSON_V
- S_TXN_LOG_ITEM_V
- S_TXN_LOG_NODE_V
- S_TXN_LOG_OPER_V

### 2.4.4 Importing the Siebel repository

Siebel Systems provides the utility imprep.ksh to import the Siebel out-of-the-box repository stored in an NT directory (this is not the repository in the DB2 for NT database). The Siebel repository contains application objects, such as views and applets.

Navigate to `d:\sea621\dbsrvr\DB2390` and edit the imprep.ksh script to contain the values shown in Table 2-5.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>SRC_USR</td>
<td>SADMIN</td>
</tr>
<tr>
<td>SRC_PSWD</td>
<td>SADMIN</td>
</tr>
<tr>
<td>SRC_TBLO</td>
<td>SIEBEL</td>
</tr>
<tr>
<td>ODBC</td>
<td>SiebServer_Siebel_390</td>
</tr>
<tr>
<td>REPOS_NAME</td>
<td>SIEBEL REPOSITORY</td>
</tr>
<tr>
<td>DBSRVR_ROOT</td>
<td>d:\sea621\dbsrvr</td>
</tr>
</tbody>
</table>
Invoke the UNIX shell by typing `sh` at the DOS prompt. Execute the script `imprep.ksh` at the shell prompt.

### 2.5 Populating the File System

A subdirectory called `files` is created automatically when you install the Siebel Database Server.

You must populate the File System directory with these file attachments after installing the Database Server, and before running the Siebel Client.

To populate the File System directory, do the following:

- Select **Start->Programs->Windows Explorer**, navigate to the `d:\sea621\dbsrvr\files` directory and copy the files to the File System `d:\siebfile` directory.
- Verify that the files are where they need to be.

### 2.6 Installing the Siebel connected client

We installed the connected client on the same machine as the other Siebel servers. This connected client points to the OS/390 database.

We selected **IBM DB2 UDB 6.1 for OS/390** on the database identification screen. The database alias is DB2DSB, the table owner is SIEBEL, and the SQLID is SSEROLE.

To see the detailed installation steps of the client, refer to *Siebel 2000 Database Implementation on OS/390 Using NT Siebel Servers*, SG24-5953. You may also follow those steps in “Installing the Siebel client” on page 113.

**Note:** When you migrate to a new platform, you need to have a new .cfg file and distribute it to all clients. Do not reuse the .cfg file from the previous platforms.
2.7 Testing the installation using Siebel Call Center

At this point in the migration process, we have created a vanilla Siebel database on OS/390 including a standard repository and seed data. Achieving this first step demonstrates that our connections to the DB2 for OS/390 work well, and that we have a Siebel-DB2 database with a repository and seed data (but no user data) in good working order.

You may now want to connect to Siebel Call Center and verify that the application connects correctly to the Siebel database on OS/390.

From the client node, log into Siebel Call Center and connect to Server as shown in Figure 2-4. Successful connection provides confirmation that the Siebel application is interacting correctly with the DB2 for OS/390 database.

![Siebel Call Center log in window](image_url)

*Figure 2-4  Log into Siebel Call Center*
Migrating Siebel data from DB2 for NT to DB2 for OS/390

This chapter describes the methods for migrating the Siebel database from a source DB2 for NT environment to a target DB2 for OS/390 environment. We consider a migration with no schema changes for this first exercise.

To move data from DB2 for NT to DB2 for OS/390, we investigated the following utilities:

- Siebel Dataexp/Dataimp and EIM utilities
- IBM DB2 Import/Export, DSNTIAUL/Load, and Cross Loader utilities

Figure 3-1 on page 32 graphically illustrates this migration.
Figure 3-1  Migrating the database from DB2 for NT to DB2 for OS/390
3.1 Preparing for data migration

At this point, we have a source environment (a production environment, for example) using DB2 for NT as the Siebel database, and a target environment (a new production environment, for example) that is using DB2 for OS/390 as the Siebel database. Now the migration of the data must be addressed.

Before moving the data, the following tasks need to be accomplished:

- Determine a list of source tables that need to be migrated
- Clean up the target database and remove seed, repository, and user data.

3.1.1 Generating a list of tables to migrate

You need to determine a list of source tables containing both system and user data that will be copied to DB2 for OS/390.

From the DB2 Command Center go to Start->Programs->DB2 For Window NT->Command Center->Script and issue the following SQL statement. Follow the Siebel Systems recommendations as described in the white paper IBM DB2 RDBMS Migration Guide Siebel 2000, which can be obtained from Siebel Systems upon request. Exclude S_DOCK_STATUS and S_DOCK_TXN_LOG from the list of tables being migrated. These tables need to be empty on the target database when data extracts are first run for remote users.

```
SELECT NAME FROM SIEBEL.S_TABLE
WHERE NAME NOT IN ('S_DOCK_STATUS','S_DOCK_TXN_LOG')
ORDER BY NAME
```

Execute the SQL statement from the DB2 Command Center and save the resulting output (Figure 3-2 on page 34) to the file Ibm_all.inp.
At this point you have created the list of table names that will be migrated from DB2 for NT to DB2 for OS/390.

You may further remove empty tables from the list of tables to migrate by checking the cardinality of the tables, assuming your statistics are up-to-date.

### 3.1.2 Cleaning up target tables

With no schema changes, migrating the data moves everything from the source to the target, including repository data, seed data, and user data. You need to delete any data you might have on the target DB2 for OS/390 database before attempting the data migration.

Go to DB2 Command Center and execute the following SQL statement (see also the white paper *IBM DB2 RDBMS Migration Guide Siebel 2000*). Be sure to exclude tables S_DOCK_STATUS and S_DOCK_TXN_LOG from the resulting table list; otherwise, the subsequent import task will fail.

```
SELECT 'DELETE FROM SIEBEL.' || NAME || ';' FROM SIEBEL.S_TABLE
WHERE NAME NOT IN ('S_DOCK_STATUS', 'S_DOCK_TXN_LOG')
ORDER BY NAME
```
In Command Center, go to **Script->Execute** to run the SQL. It will generate the output shown in Figure 3-3.

```
delete from siebel.EAI_ACCOUNT_IF;
delete from siebel.EAI_ADDRESS_IF;
delete from siebel.EAI_CONTACT_IF;
delete from siebel.EAI_PROD_IF;
delete from siebel.EIM_HCCNT_DTL;
delete from siebel.EIM_HCCNT_SRC;
delete from siebel.EIM_ACCOUNT;
delete from siebel.EIM_ACCOUNT1;
delete from siebel.EIM_ACCOUNT2;
```

*Figure 3-3  SQL delete statement*

Save the results to del_all_s_tabs.sql, then ship the file to OS/390 and execute from DB2 Spufi. At this point you have cleaned up the seed data and repository data contained in the target Siebel database tables on OS/390.

The target tables are now clean and empty of any Siebel data on OS/390. You are now ready to move the data.

### 3.2 Moving data from DB2 for NT to DB2 for OS/390

Data can be moved in different ways from the source to the target Siebel database. We describe the following techniques:

- **Siebel utilities**
  - Dataexp/Dataimp
  - EIM

- **IBM DB2 utilities**
  - DB2 Import/Export
  - DB2 DSNTIaul/Load
  - DB2 Family Cross Loader function
The premise for using the IBM DB2 data movement utilities is that the migrated target Siebel database server will be an exact replica of the Siebel source database server. All Siebel database objects and data must be migrated from the source system to the target system. Based on this premise, we can assume that all inter-table relationships (such as the Referential Integrity) are maintained. Any customizing must be performed prior to or following the migration, not during the migration process.

3.3 Siebel Dataexp/Dataimp - no schema changes

The Siebel Dataexp and Dataimp utilities execute on the NT platform as shown in Figure 3-4.

With no schema changes, the Dataexp/Dataimp migration process includes the following:

- Export data from DB2 for NT database including seed data, repository data and user data. The export job creates a .dat output file.
- Import data from the .dat output file created by the export, into DB2 for OS/390 via SQL inserts.
- Validate the migration.

3.3.1 Exporting data with Dataexp

You can find dataexp.exe by navigating to the d:\sea621\siebsrvr\bin directory. Obtain a description of the parameters for exporting data by entering dataexp/? at the command line as shown in Figure 3-5.
Execute this module by entering the following command:

d:\sea621\siebsrvr\bin\dataexp /u SIEBEL /p SIEBEL /c SiebSrvr_siebel /d SIEBEL /f d:\Mov_dat/ibm_all.dat /i d:\Mov_dat/ibm_all.inp

This job creates the output file ibm_all.dat. This file will be used as input by the dataimp.exe module to import the data to the OS/390.

3.3.2 Importing data with Dataimp

You can find dataimp.exe by navigating to the d:\sea621\siebsrvr\bin directory. Obtain a description of the parameters used in previous commands by navigating to the d:\sea621\siebsrvr\bin directory and entering dataimp/? at the command line, as shown in Figure 3-6 on page 37.
Execute this module by entering the following command:

d:\sea621\siebsrvr\bin\dataimp /u SIEBEL /p SIEBEL /c DB2DSB /d SIEBEL /f d:\Mov_dat\ibm_all.dat

At the end of this job you have imported the repository, the seed data, and all user data to the target Siebel database on OS/390.

### 3.3.3 Validating the migration

To validate the data migration, connect from the Siebel client application to OS/390. Navigate to Siebel Call Center and make sure that the migrated user data exists there. You should be able to see the data entered in “Entering data using Siebel Call Center” on page 121.

### 3.4 Siebel EIM

Siebel Enterprise Integration Manager (EIM) is the Siebel-sanctioned technique to import and export data external to the Siebel system and is typically thought of as the technique to integrate data to and from legacy systems. This is because using EIM ensures that Siebel referential integrity is maintained.

It is also an option for migrating Siebel data from one RDBMS, or hardware platform, to another. In the source RDBMS system, we use the EIM export function to move the Siebel base tables into its EIM tables, unload these EIM tables and reload them into the target system’s EIM tables, and then use EIM import utility to move them into the target system’s tables.

To be able to use EIM, you must first find the EIM tables that map the base tables. In our test, we only mapped a few base tables, but for a migration of all data to the target RDBMS, you need to map all the system’s tables to EIM.

There is a caveat if this is being done. At this time, the Siebel EIM tables do not support all the base tables that can be created, so if you have chosen Siebel functions that result in the use of these tables, a manual intervention will be necessary. This is a dynamic situation and each release can affect which, if any, tables are not supported. If you encounter a situation like this, there are reasonable techniques to handle it, by writing a Siebel VB routine or using the Enterprise Application Interface (EAI) to populate the tables in question.

The migration of your tables with EIM involves four steps as shown in Figure 3-7—all utilities run on the NT platform:

1. EIM Export
2. Dataexp
3.4.1 EIM export

To export a table with EIM, do the following:

- Identify the base tables that will be migrated. For testing purposes, we decided to migrate one table S_org_ext. But because of the relationships of this table to other tables, we had to migrate the following five tables:
  - S_org_ext
  - S_acct_postn
  - S_addr_org
  - S_org_indust
  - S_indust

- The next step is to identify the target interface tables related to the five tables we want to migrate. We searched for the base table names in the Interface Tables Reference Guide.

Another way of searching for EIM tables is to run the following SQL statement using the relationship tables. We searched on S_acct_postn relationship as shown in Figure 3-8:
Migrating Siebel Database from DB2/Oracle for NT to DB2 for OS/390

40

Figure 3-8  SQL statement to search for S_acct_postn interface tables

The result of this SQL statement is shown in Figure 3-9:

Figure 3-9  Resulting EIM tables for S_acct_postn

Focusing on the relationship tables (S_acct_postn, and S_org_indust), we identified the following interface tables:

- EIM_account (populates S_org_ext, S_acct_postn, S_addr_org)
- EIM_account1 (populates S_org_indust)
- EIM_indust (populates S_indust)

We didn’t need to export position data because we used the standard positions provided in the seed data.

Figure 3-10 shows—for the base table S_indust—all the columns in the interface table EIM_industry that have to be populated.
To populate the EIM input parameter file for the first time, we copied the default.ifb (d:\sea621\siebsrvr\admin) file into a file we called export.ifb. We then updated this file with our own values, specifying what we want to do (export), and the interface tables to use for this process (EIM_industry, EIM_account, EIM_account1), as shown in Figure 3-11.

The next step is to invoke the EIM server task:
- From the client workstation, get into the Siebel Call Center.
- Go to Screens->Server Administration->Server->Server Tasks->Click as shown in Figure 3-12.
On the Server Task screen select New.
In the Component drop-down list, select Enterprise Integration Mgr.
Select Pick.
On the server task screen, select Parameters.
Change the configuration file parameter to contain export.ifb.
Close the window.

To start the EIM task, click Start. The Siebel application generates a task number for the EIM task. This number is used to check the results of the export.

When the EIM export task completes, the status field displays the Completed status as shown in Figure 3-13.

At this point EIM export is done and the data is moved from the base tables into the EIM interface tables. Nevertheless, we have to check the number of records we have exported. Please note that this will be more than the number
Chapter 3. Migrating Siebel data from DB2 for NT to DB2 for OS/390

of accounts in the system as an additional record is exported for each relationship.

Navigate the d:\sea621\siebsrvr\log directory and select the log file EIM_3082 (task number). Review the log file for any errors, as shown in Figure 3-14. It shows how many records have been exported to the interface table for each of our specified tables, EIM_INDUSTRY, EIM_ACCOUNT and EIM_ACCOUNT1.

<table>
<thead>
<tr>
<th>Table</th>
<th>Base Table</th>
<th>Rows Exported</th>
<th>Total Time (s)</th>
<th>Row Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>EIM_INDUSTRY</td>
<td>S_INDUSTRY</td>
<td>4</td>
<td>0.16s</td>
<td>0.04s</td>
</tr>
<tr>
<td>EIM_ACCOUNT</td>
<td>S_ACCOUNT</td>
<td>14</td>
<td>0.43s</td>
<td>0.03s</td>
</tr>
<tr>
<td>EIM_ACCOUNT</td>
<td>S_IDOGR_DEV</td>
<td>14</td>
<td>0.27s</td>
<td>0.02s</td>
</tr>
<tr>
<td>EIM_ACCOUNT</td>
<td>S_ORG_REL</td>
<td>8</td>
<td>0.19s</td>
<td>0.02s</td>
</tr>
<tr>
<td>EIM_ACCOUNT</td>
<td>S_ACCOUNT</td>
<td>11</td>
<td>0.18s</td>
<td>0.02s</td>
</tr>
<tr>
<td>EIM_ACCOUNT</td>
<td>S_ACCOUNT</td>
<td>29</td>
<td>2.1s</td>
<td>0.09s</td>
</tr>
<tr>
<td>EIM_ACCOUNT</td>
<td>S_ACCOUNT</td>
<td>0</td>
<td>0.13s</td>
<td>0.05s</td>
</tr>
<tr>
<td>EIM_ACCOUNT</td>
<td>S_ACCOUNT</td>
<td>0</td>
<td>0.10s</td>
<td>0.05s</td>
</tr>
<tr>
<td>EIM_ACCOUNT</td>
<td>S_ACCOUNT</td>
<td>0</td>
<td>0.15s</td>
<td>0.05s</td>
</tr>
<tr>
<td>EIM_ACCOUNT</td>
<td>S_ACCOUNT</td>
<td>0</td>
<td>0.09s</td>
<td>0.05s</td>
</tr>
<tr>
<td>EIM_ACCOUNT</td>
<td>S_ACCOUNT</td>
<td>0</td>
<td>0.10s</td>
<td>0.05s</td>
</tr>
<tr>
<td>EIM_ACCOUNT</td>
<td>S_ACCOUNT</td>
<td>0</td>
<td>0.11s</td>
<td>0.05s</td>
</tr>
<tr>
<td>EIM_ACCOUNT</td>
<td>S_ACCOUNT</td>
<td>0</td>
<td>0.12s</td>
<td>0.05s</td>
</tr>
<tr>
<td>EIM_ACCOUNT</td>
<td>S_ACCOUNT</td>
<td>0</td>
<td>0.13s</td>
<td>0.05s</td>
</tr>
<tr>
<td>EIM_ACCOUNT</td>
<td>S_ACCOUNT</td>
<td>0</td>
<td>0.13s</td>
<td>0.05s</td>
</tr>
<tr>
<td>EIM_ACCOUNT</td>
<td>S_ACCOUNT</td>
<td>0</td>
<td>0.14s</td>
<td>0.05s</td>
</tr>
<tr>
<td>EIM_ACCOUNT</td>
<td>S_ACCOUNT</td>
<td>0</td>
<td>0.15s</td>
<td>0.05s</td>
</tr>
<tr>
<td>EIM_ACCOUNT</td>
<td>S_ACCOUNT</td>
<td>0</td>
<td>0.16s</td>
<td>0.05s</td>
</tr>
<tr>
<td>EIM_ACCOUNT</td>
<td>S_ACCOUNT</td>
<td>0</td>
<td>0.17s</td>
<td>0.05s</td>
</tr>
<tr>
<td>EIM_ACCOUNT</td>
<td>S_ACCOUNT</td>
<td>0</td>
<td>0.18s</td>
<td>0.05s</td>
</tr>
<tr>
<td>EIM_ACCOUNT</td>
<td>S_ACCOUNT</td>
<td>0</td>
<td>0.19s</td>
<td>0.05s</td>
</tr>
<tr>
<td>EIM_ACCOUNT</td>
<td>S_ACCOUNT</td>
<td>0</td>
<td>0.20s</td>
<td>0.05s</td>
</tr>
<tr>
<td>EIM_ACCOUNT</td>
<td>S_ACCOUNT</td>
<td>0</td>
<td>0.21s</td>
<td>0.05s</td>
</tr>
<tr>
<td>EIM_ACCOUNT</td>
<td>S_ACCOUNT</td>
<td>0</td>
<td>0.22s</td>
<td>0.05s</td>
</tr>
<tr>
<td>EIM_ACCOUNT</td>
<td>S_ACCOUNT</td>
<td>0</td>
<td>0.23s</td>
<td>0.05s</td>
</tr>
<tr>
<td>EIM_ACCOUNT</td>
<td>S_ACCOUNT</td>
<td>0</td>
<td>0.24s</td>
<td>0.05s</td>
</tr>
<tr>
<td>EIM_ACCOUNT</td>
<td>S_ACCOUNT</td>
<td>0</td>
<td>0.25s</td>
<td>0.05s</td>
</tr>
<tr>
<td>EIM_ACCOUNT</td>
<td>S_ACCOUNT</td>
<td>0</td>
<td>0.26s</td>
<td>0.05s</td>
</tr>
<tr>
<td>EIM_ACCOUNT</td>
<td>S_ACCOUNT</td>
<td>0</td>
<td>0.27s</td>
<td>0.05s</td>
</tr>
<tr>
<td>EIM_ACCOUNT</td>
<td>S_ACCOUNT</td>
<td>0</td>
<td>0.28s</td>
<td>0.05s</td>
</tr>
<tr>
<td>EIM_ACCOUNT</td>
<td>S_ACCOUNT</td>
<td>0</td>
<td>0.29s</td>
<td>0.05s</td>
</tr>
<tr>
<td>EIM_ACCOUNT</td>
<td>S_ACCOUNT</td>
<td>0</td>
<td>0.30s</td>
<td>0.05s</td>
</tr>
<tr>
<td>EIM_ACCOUNT</td>
<td>S_ACCOUNT</td>
<td>0</td>
<td>0.31s</td>
<td>0.05s</td>
</tr>
<tr>
<td>EIM_ACCOUNT</td>
<td>S_ACCOUNT</td>
<td>0</td>
<td>0.32s</td>
<td>0.05s</td>
</tr>
<tr>
<td>EIM_ACCOUNT</td>
<td>S_ACCOUNT</td>
<td>0</td>
<td>0.33s</td>
<td>0.05s</td>
</tr>
<tr>
<td>EIM_ACCOUNT</td>
<td>S_ACCOUNT</td>
<td>0</td>
<td>0.34s</td>
<td>0.05s</td>
</tr>
<tr>
<td>EIM_ACCOUNT</td>
<td>S_ACCOUNT</td>
<td>0</td>
<td>0.35s</td>
<td>0.05s</td>
</tr>
<tr>
<td>EIM_ACCOUNT</td>
<td>S_ACCOUNT</td>
<td>0</td>
<td>0.36s</td>
<td>0.05s</td>
</tr>
<tr>
<td>EIM_ACCOUNT</td>
<td>S_ACCOUNT</td>
<td>0</td>
<td>0.37s</td>
<td>0.05s</td>
</tr>
<tr>
<td>EIM_ACCOUNT</td>
<td>S_ACCOUNT</td>
<td>0</td>
<td>0.38s</td>
<td>0.05s</td>
</tr>
<tr>
<td>EIM_ACCOUNT</td>
<td>S_ACCOUNT</td>
<td>0</td>
<td>0.39s</td>
<td>0.05s</td>
</tr>
<tr>
<td>EIM_ACCOUNT</td>
<td>S_ACCOUNT</td>
<td>0</td>
<td>0.40s</td>
<td>0.05s</td>
</tr>
<tr>
<td>EIM_ACCOUNT</td>
<td>S_ACCOUNT</td>
<td>0</td>
<td>0.41s</td>
<td>0.05s</td>
</tr>
<tr>
<td>EIM_ACCOUNT</td>
<td>S_ACCOUNT</td>
<td>0</td>
<td>0.42s</td>
<td>0.05s</td>
</tr>
<tr>
<td>EIM_ACCOUNT</td>
<td>S_ACCOUNT</td>
<td>0</td>
<td>0.43s</td>
<td>0.05s</td>
</tr>
<tr>
<td>EIM_ACCOUNT</td>
<td>S_ACCOUNT</td>
<td>0</td>
<td>0.44s</td>
<td>0.05s</td>
</tr>
<tr>
<td>EIM_ACCOUNT</td>
<td>S_ACCOUNT</td>
<td>0</td>
<td>0.45s</td>
<td>0.05s</td>
</tr>
<tr>
<td>EIM_ACCOUNT</td>
<td>S_ACCOUNT</td>
<td>0</td>
<td>0.46s</td>
<td>0.05s</td>
</tr>
<tr>
<td>EIM_ACCOUNT</td>
<td>S_ACCOUNT</td>
<td>0</td>
<td>0.47s</td>
<td>0.05s</td>
</tr>
<tr>
<td>EIM_ACCOUNT</td>
<td>S_ACCOUNT</td>
<td>0</td>
<td>0.48s</td>
<td>0.05s</td>
</tr>
<tr>
<td>EIM_ACCOUNT</td>
<td>S_ACCOUNT</td>
<td>0</td>
<td>0.49s</td>
<td>0.05s</td>
</tr>
<tr>
<td>EIM_ACCOUNT</td>
<td>S_ACCOUNT</td>
<td>0</td>
<td>0.50s</td>
<td>0.05s</td>
</tr>
<tr>
<td>EIM_ACCOUNT</td>
<td>S_ACCOUNT</td>
<td>0</td>
<td>0.51s</td>
<td>0.05s</td>
</tr>
<tr>
<td>EIM_ACCOUNT</td>
<td>S_ACCOUNT</td>
<td>0</td>
<td>0.52s</td>
<td>0.05s</td>
</tr>
<tr>
<td>EIM_ACCOUNT</td>
<td>S_ACCOUNT</td>
<td>0</td>
<td>0.53s</td>
<td>0.05s</td>
</tr>
<tr>
<td>EIM_ACCOUNT</td>
<td>S_ACCOUNT</td>
<td>0</td>
<td>0.54s</td>
<td>0.05s</td>
</tr>
<tr>
<td>EIM_ACCOUNT</td>
<td>S_ACCOUNT</td>
<td>0</td>
<td>0.55s</td>
<td>0.05s</td>
</tr>
<tr>
<td>EIM_ACCOUNT</td>
<td>S_ACCOUNT</td>
<td>0</td>
<td>0.56s</td>
<td>0.05s</td>
</tr>
<tr>
<td>EIM_ACCOUNT</td>
<td>S_ACCOUNT</td>
<td>0</td>
<td>0.57s</td>
<td>0.05s</td>
</tr>
<tr>
<td>EIM_ACCOUNT</td>
<td>S_ACCOUNT</td>
<td>0</td>
<td>0.58s</td>
<td>0.05s</td>
</tr>
<tr>
<td>EIM_ACCOUNT</td>
<td>S_ACCOUNT</td>
<td>0</td>
<td>0.59s</td>
<td>0.05s</td>
</tr>
<tr>
<td>EIM_ACCOUNT</td>
<td>S_ACCOUNT</td>
<td>0</td>
<td>0.60s</td>
<td>0.05s</td>
</tr>
<tr>
<td>EIM_ACCOUNT</td>
<td>S_ACCOUNT</td>
<td>0</td>
<td>0.61s</td>
<td>0.05s</td>
</tr>
</tbody>
</table>

Figure 3-14  Log file for EIM export

3.4.2 Dataexp

You are now ready to export the EIM interface tables using the dataexp.exe utility. Dataexp generates a flat file named ibm_1.dat.

Execute the dataexp utility from the DOS prompt, as follows:

d:\sea621\siebsrvr\bin\dataexp /u sadmin /p sadmin /c SiebSrvr_siebel /d SIEBEL /f d:\Mov_dat\ibm_1.dat /i d:\Mov_dat\ibm_1inp

After execution, review the log dataexp.log file for any errors. Figure 3-15 on page 44 shows our dataexp.log file. It confirms that we exported 3 tables and 67 rows.
3.4.3 Dataimp

We now imported the content of ibm_1.dat in the EIM interface tables on the OS/390 environment. We did this in a client/server mode from the NT.

Execute (on the NT) the dataimp.exe utility from the DOS prompt, as follows:

```
d:\sea621\siebsrvr\bin\dataimp /u sadmin /p sadmin /c DB2DSB /d SIEBEL /f d:\Mov_dat\ibm_1.bat
```

Note that the /c parameter is pointing to the OS/390 ODBC data source.

Review the log file dataimp.log for any errors. Figure 3-16 on page 44 shows our dataimp.log output.
At the end of the Dataimp step we had moved the data into the EIM interface tables on the OS/390.

### 3.4.4 EIM Import

We ran the EIM Import utility on the NT. The utility opens an ODBC connection to OS/390 where it reads the EIM interface tables (populated by Dataimp) and inserts them into the target DB2 tables.

- To populate the EIM input parameter file for the first time, we copied the default.ifb (d:\sea621\siebsrvr\admin) file into a file we named import.ifb. We then updated this file with our own values, specifying what we want to do (import), and the interface tables to use for this process (EIM_industry, EIM_account, EIM_account1). The edited import.ifb file is displayed in Figure 3-17.

![The edited import.ifb file](image)

**Figure 3-17 The edited import.ifb file**

- The next step is to invoke the EIM server task:
  - From the client workstation, get into the Siebel Call Center.
  - Go to **Screens->Server Administration->Server->Server Tasks->Click**.
  - On the Server Task screen select **New**.
  - In the Component drop-down list select **Enterprise Integration Mgr.**
  - Select **Pick**.
  - On the server task screen select **Parameters**.
  - Change the Configuration file parameter to contain import.ifb.
  - Close the window.
To start the EIM task, click **Start**. The Siebel application generates a task number for the EIM task. This number is used to check the results of the import.

When the EIM import task completes, the status field displays the Completed status.

You need now to confirm that data has been moved correctly by doing the following:
- Check for error messages in the EIM log file - EIM_3085 (the name of the EIM log file contains the EIM task number).
- Using the Siebel Call Center, check whether sample records have been imported correctly into OS/390.

We specifically checked the Accounts screen to make sure that all our accounts were there. We also checked the addresses we imported.

### 3.5 DB2 for NT Import/Export

We tested the DB2 for NT Import/Export utilities for migrating a couple of EIM tables. We encountered a condition where the DB2 Import utility failed because the schema was different on the target platform. The target EIM table had a column, MS_IDENT, which did not exist on the source EIM table. This column was defined as an IDENTITY COLUMN, so there really was no need to move data, but the DB2 Import utility did not tolerate the mismatch.

The DB2 Import utility could have been customized to bypass this problem by specifying the columns to be moved, but the large number of columns needed to be specified eliminated this option.

Another alternative would have been to create views to include all columns except the IDENT column and load the data using the views.
3.6 DB2 for OS/390 DSNTIAUL and LOAD

The following database migration scenario was tested as a joint Siebel-IBM effort in Siebel premises, using Siebel utilities to extract and recreate database objects, and using IBM DB2 for OS/390 utilities to extract and reload the data. As a result of the tests, the solution was reported as quite efficient.

The scenario described in Figure 3-18 uses the DB2 for OS/390 DSNTIAUL and LOAD utilities to move the data. The LOAD utility is especially efficient in loading large volumes of data.

The DB2 for OS/390 sample utility program, DSNTIAUL, running on OS/390 and using an implicit remote DRDA connection to DB2 for NT is used to extract data from DB2 for NT tables and store it in files on OS/390. DSNTIAUL extracts data from DB2 for NT and stores the unloaded data in a "loadable" format acceptable to the DB2 for OS/390 LOAD utility. DSNTIAUL is also used to generate the LOAD utility control files while unloading the DB2 for NT data.

The steps to migrate from a Siebel DB2 for NT database server to a DB2 for OS/390 database server are as follows:

1. Define the source DB2 for NT as a DRDA server to DB2 for OS/390, giving it an RDBNAME which can be used by DSNTIAUL. To do this, update the DB2 for OS/390 Communications Database to point to DB2 for NT via TCP/IP.
   - Update the SYSIBM.LOCATIONS LOCATION column with the location name of the source DB2 for NT, the LINKNAME column with a value identifying a row in the SYSIBM.IPNAMES table, and the PORT column.

---

Figure 3-18   Migrating Siebel data with DB2 DSNTIAUL and LOAD utilities

The steps to migrate from a Siebel DB2 for NT database server to a DB2 for OS/390 database server are as follows:

1. Define the source DB2 for NT as a DRDA server to DB2 for OS/390, giving it an RDBNAME which can be used by DSNTIAUL. To do this, update the DB2 for OS/390 Communications Database to point to DB2 for NT via TCP/IP.
   - Update the SYSIBM.LOCATIONS LOCATION column with the location name of the source DB2 for NT, the LINKNAME column with a value identifying a row in the SYSIBM.IPNAMES table, and the PORT column.
with the port number associated with the source DB2 for NT in your configuration (50000 in our case).

<table>
<thead>
<tr>
<th>SYSSBM.SYSLOCATIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOCATION</td>
</tr>
<tr>
<td>SIEBELDB</td>
</tr>
</tbody>
</table>

– Update the SYSSBM.IPNNAMES LINKNAME column with the value used in the SYSSBM.LOCATIONS LINKNAME column, the IPADDR column with the IP address of the source DB2 for NT server, the SECURITY_OUT column with P indicating a user ID and password will be sent during the connect process. With the P option, you need a SYSSBM.USERNAMES table for user ID/password resolution. In the USERNAMES column, specify O to indicate outbound translation.

<table>
<thead>
<tr>
<th>SYSSBM.IPNNAMES</th>
</tr>
</thead>
<tbody>
<tr>
<td>LINKNAME</td>
</tr>
<tr>
<td>SIEBELNK</td>
</tr>
</tbody>
</table>

– Update SYSSBM.USERNAMES column TYPE to O for outbound translation, leave AUTHID blank to translate all user IDs, set the column LINKNAME to the same value used in the SYSSBM.LOCATIONS LINKNAME column. For the NEWAUTHID and PASSWORD columns specify the user ID and password of a user that has select access on all the Siebel tables.

<table>
<thead>
<tr>
<th>SYSSBM.USERNAMES</th>
</tr>
</thead>
<tbody>
<tr>
<td>TYPE</td>
</tr>
<tr>
<td>O</td>
</tr>
</tbody>
</table>

2. Use the installation job DSNTEJ2A to recompile, assemble, and linkedit DSNTIAUL. You need to modify the bind step since you must bind DSNTIAUL's DBRM using the CURRENTSERVER bind option specifying the location name used in the LOCATION column in the SYSSBM.LOCATIONS table. A plan and package is required to run DSNTIAUL, accessing the source DB2 for NT server.

– To build a plan and package for DSNTIAUL, you cannot use certain bind options. When binding the DSNTIAUL package to DB2 for NT, you can
only specify the following options: CURRENTDATA NO, ISOLATION CS or
UR, and VALIDATE BIND. Do not specify SQLERROR CONTINUE. Bind a
plan and specify a package list where the PKLIST includes ".DSNTIAUL.*",
meaning any package in the DSNTIAUL collection at any location. Be sure
to specify the CURRENTSERVER keyword in the bind of the plan so that
an automatic connection is established when the application begins to
execute and connects to DB2 for OS/390.

3. Build JCL to run one or more DSNTIAUL jobs using the plan created
previously. Refer to Appendix E, “DB2 DSNTIAUL and LOAD jobs” on
page 159, to see an example. Run DSNTIAUL to select all data from all tables
plus creating load control files for all tables. You can run multiple DSNTIAUL
jobs to extract data in parallel.
   – You can use the TSO DSN command processor to run DSNTIAUL. Look at
     the execute step of DSNTEJ2A to see how this is done.
   – Creating the JCL jobs to run DSNTIAUL against all the Siebel tables can
     be very time consuming, so writing a REXX exec to create a set of
     DSNTIAUL jobs is recommended. “REXX EXEC to generate DSNTIAUL
     unload jobs” on page 162 and “REXX EXEC to generate DB2 LOAD jobs”
     on page 165, show two REXX execs that can be used as samples to
     generate the DSNTIAUL unload jobs and the DB2 LOAD jobs.
   – The SYSPUNCH DD card is for the LOAD control cards and SYSREC00 is
     for the unloaded data, with SYSIN being just the fully qualified name of the
     table on input. One of the issues with this unload is that the file records are
     all padded out to full column lengths including the max column length of a
     long varchar which means there is a lot of null data in these files but they
     need to be there. By using an SMS storage group with stripping across,
     say, 8 disks, and compression, can reduce the DASD requirement.

4. Build JCL to run the DB2 for OS/390 LOAD utility to load the data extracted by
DSNTIAUL. Refer to Appendix E, “DB2 DSNTIAUL and LOAD jobs” on
page 159, to see an example of this process.

5. Run RUNSTATS to populate the statistics column for the Siebel tables.

6. If any triggers were created on the source database they have to be recreated
on the target database using the normal Siebel Administration facilities.

7. Collect a count of all Siebel tables, indexes, triggers, and row counts for each
table. Compare there results with the counts extracted from DB2 for NT.

8. Configure the Siebel clients and Application servers to point to the new Siebel
database server.

9. Validate the schema.
This is the outline of the process required to migrate a DB2 for NT server to a DB2 for OS/390 server. Additional planning is required to put this in production. Authorization, backup and restore, disaster recovery, and performance tuning all have to be examined and merged into the current DB2 for OS/390 production processes.

Timings

It took 8 to 12 hours for the Siebel DB2 for OS/390 DBA to move 40 GB of data from DB2 for NT using DSNTIAUL running on OS/390. This was an early attempt and times have improved since. The additional time for the first move was because the operation was not optimized for DASD space usage. We estimate that this time can be cut to 4 to 6 hours with sufficient SMS-managed DASD.

To unload the approximately 1800 Siebel tables, the DSNTIAUL jobs were split into 18 physical jobs. DSNTIAUL can only take SYSREC0 - SYSREC99, so divide the number of tables by 100 for the number of jobs.

In our case, we run four job streams at a time mainly due to the amount of DASD available for DB2’s usage. You will have to adjust the number of parallel streams based on the amount of DASD available for the SORTOUT, SYSUT1, SORTWK01- SORTWK08 datasets. You can experiment to see how many jobs can be run in parallel or you can go through the exercise of calculating the exact amount of DASD required using the formulas in the DB2 manuals. There is no simple formula since you have to take into account the number of rows, the number of indexes, and the sizes of indexes. The JCL attempts to handle the largest table so for most tables the allocation amounts may seem high but you have to be ready for peak demands if you want to run this process unattended.

To reload the data into DB2 for OS/390 after the unload, figure between 6 to 12 hours for the 40 GB. Run 4 jobs at a time that have multiple tables per job. They may clash on table space, DASD, core, etc. They need to be monitored for continuous processing. Again, having enough SMS-managed DASD helps speed things up. The loads include “resume yes”, indexes, and a lot of buffers (250 on input data sets and works and sorts); also specify external sorts (24). The total time includes parallel RUNSTAT jobs (4) after all the loads, but does not include a COPY.

3.7 DB2 family cross-loader function

The DB2 family cross-loader function is an alternative to the scenario described in the previous section, and available only in the DB2 for OS/390 V7 environments. It is a powerful data migration tool for the DB2 family.
With DB2 for OS/390 Version 7, the DB2 LOAD utility is enhanced to support the use of SQL SELECT to produce input directly into LOAD. This enhancement delivers a DB2 family cross-loader function. With DB2 for OS/390 Version 7, you can use any DRDA-compliant database server as a data input source for populating your tables in DB2 for OS/390. Your input can even come from non-DB2 data sources by using DataJoiner for access to data from sources as diverse as Oracle and Sybase, as well as the entire DB2 family of database servers.

Siebel 2000 is not currently supporting DB2 for OS/390 V7, but is likely to do it at a later time. The DB2 family cross-loader function will be available to Siebel Systems as soon as it supports DB2 for OS/390 V7.

Chapter 4. Migrating Siebel data from Oracle for NT to DB2 for NT

This chapter describes the methods we used for migrating the Siebel database from a source Oracle for NT environment to a target DB2 for NT environment, shown in Figure 4-1.

Figure 4-1  Migrating Siebel data from Oracle for NT to DB2 for NT
4.1 Test configuration

We ran the Oracle for NT to DB2 for NT migration tests on a configuration we set up at the IBM Hursley laboratory in the UK.

We used Siebel 2000 V6.2.1 with Fix Pack 3sa (Fix Pack 6s was not available at the time of testing).

1. First, we created a Siebel source environment on NT supporting an Oracle for NT V8.1.6 Siebel database, using the same methodology as in Appendix A, “Implementing the Siebel source environment on NT” on page 89. We populated the source Oracle database with SQL Anywhere sample data. (See “Migrating the Siebel sample database” on page 124).

2. Next, we created a Siebel target environment on NT supporting the DB2 for NT V6 Siebel database, again using the same methodology as in Appendix A, “Implementing the Siebel source environment on NT” on page 89.

3. Finally, we moved the data from the source Oracle for NT to the target DB2 for NT environment.

We chose the following naming conventions for the source Oracle for NT setup:

<table>
<thead>
<tr>
<th>NT account/psw</th>
<th>SADMIN/SADMIN and SIEBEL/SIEBEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enterprise server name</td>
<td>SiebelOracle</td>
</tr>
<tr>
<td>Database SSID</td>
<td>SANDPIT</td>
</tr>
<tr>
<td>Table owner</td>
<td>SIEBEL</td>
</tr>
<tr>
<td>Database account login/psw</td>
<td>SADMIN/SADMIN</td>
</tr>
<tr>
<td>ODBC data source name</td>
<td>Siebsrvr_SiebelOracle</td>
</tr>
</tbody>
</table>

We chose the following naming conventions for the target DB2 for NT setup:

<table>
<thead>
<tr>
<th>NT account/password</th>
<th>SADMIN/SADMIN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database account/password</td>
<td>SADMIN/SADMIN</td>
</tr>
<tr>
<td>Enterprise server name</td>
<td>SiebelDB2</td>
</tr>
<tr>
<td>Table owner</td>
<td>SIEBEL</td>
</tr>
<tr>
<td>Database alias</td>
<td>siebeldb</td>
</tr>
<tr>
<td>Data source name</td>
<td>Siebsrvr_SiebelDB2</td>
</tr>
</tbody>
</table>
4.2 Migrating Siebel data from Oracle for NT to DB2 for NT

We investigated the following methods using Siebel utilities:

- The Dataexp/Dataimp method with schema changes
- The Dataexp/Dataimp method with no schema changes
- The EIM method

4.2.1 Siebel Dataexp/Dataimp - with schema changes

This method handles schema differences between the source and target environments. Figure 4-2 describes migrating Siebel data from Oracle to DB2 using the Dataexp and Dataimp utilities.

![Figure 4-2](image)

The procedure includes the following:

- Generating a list of tables to migrate
- Cleaning up target tables
- Renaming the repository at the target
- Moving the repository and synchronizing the schema
- Recompiling the SRF file
- Exporting data with Dataexp
- Importing data with Dataimp
- Validating the migration
Generating a list of tables to migrate

Before migrating the data, we need to determine which tables need to be migrated. We will migrate all tables except the repository, which will be handled in "Moving the repository and synchronizing the schema" on page 61.

From the Windows desktop, go to **Start->Programs->Oracle->Application Development->SQL Plus** and issue the SQL statement shown in Figure 4-3.

```
SQL>
select name
from siebel.s_table
where (type like 'Data%' or type = 'Extension')
and name not in ('S_DOCK_STATUS','S_DOCK_TXN_LOG')
order by name
```

*Figure 4-3  SQL statement to generate table names*

The output of this SQL statement will give us a list of table names we will want to migrate and will be used as input for Dataexp. The output is shown in Figure 4-4.

```
NAME
-----------------------
S_WF_COND_VAL
S_WF_PROC_FLOW
S_WF_PROC_PROP
S_WF_PROP_VAL
S_WF_STEP
S_WF_STEP_ARG
S_WF_STEP_BRANCH
S_WF_STEP_INST
S_WF_STEP_PEND
S_WANTY_CURG
S_WANTY_SRV_ORG

NAME
-----------------------
S_ZIPCODE
```

*Figure 4-4  SQL statement output with table names*
Drop the comments from the output file and save only the table names into a file named `ora_all_out.inp`, for example. This table list is used as input to the Dataexp utility later.

### Cleaning up target tables

We need to make sure that there is no data in the target tables before we start migrating the data. We want to avoid running into duplicate data issues.

Go to SQL Plus and build the SQL statement shown in Figure 4-5.

```
SQL> line 6 truncated.
  1  select 'delete from siebel."||name"';
  2      from siebel.s_table
  3      where (type like 'Data%' or type = 'Extension')
  4      and name not in ('S Dock_Status','S Dock_TXN_LOG')
  5      order by name

Figure 4-5  SQL statement to generate SQL DELETE statements
```

Running this SQL statement produces the output shown in Figure 4-6 on page 58. It builds the SQL DELETE statements required to clean up the target tables.
Consolidate the output and save it as del_data_run.inp, for example, and execute it in DB2 Command Center before you run Dataimp. See Figure 4-7 on page 59.
Figure 4-7 Executing the DELETE statements from the DB2 Command Center

Figure 4-8 on page 60 shows the result of the execution of the SQL DELETE statements. You should expect to see messages informing you that there are no rows in the tables because we are not using most of the Siebel business components.
At the end of the execution, the target tables will be cleaned up.

**Renaming the repository**

When the repository gets moved to the target, we create a situation where there are two repositories (the target and the moved one) with the same name, which is a problem. To avoid this problem, we rename the repository at the target.

To rename the repository, we use Siebel Tools from the client workstation and change the name Siebel Repository to *OLD Siebel Repository*. See Figure 4-9 on page 61.
Chapter 4. Migrating Siebel data from Oracle for NT to DB2 for NT

Figure 4-9  Using Siebel Tools to rename the repository

Moving the repository and synchronizing the schema
Dev2prod will move the repository and synchronize the logical and physical schema at the target environment.

Navigate to E:\sea621\db2udb\dbsrvr\db2udb\dev2prod.ksh. Edit and update it according to your environment settings, as shown in Figure 4-10 on page 62.
# do not remove quotes
SRC_USR=SADMIN
SRC_PWD=SADMIN
SRC_TDDO=SIEBEL
SRC_TDDO_PWD=SIEBEL
SRC_REPO="Siebel Repository"
SRC_ODBC=SiebSrvr_siebe1oracle
TGT_USR=SADMIN
TGT_PWD=SADMIN
TGT_TDDO=SIEBEL
TGT_TDDO_PWD=SIEBEL
TGT_REPO="Siebel Repository"
TGT_ODBC=SiebSrvr_siebe2db

DATA_AREA="SIEBEL_AK"
INDEX_AREA="SIEBEL_IDX"
LONG_AREA="SIEBEL_AWL"
DB4OK_AREA="SIEBEL_4OK"
DATABASE_PLATFORM=DB2udb

DB_UNICODE_FLG=H

SIEBEL_HOME=E:/seaA21/db2udb/siebSrvr
	typeset -x SIEBEL_HOME

# Do Not Edit Below this line
#
# =---------------------------------------------------------------
#
# case $DATABASE_PLATFORM in
# Informix|MySQL|Server|Oracle|Sybase|DB2udb)
#  ::
#  Unspecified("")
#  echo Invalid Value for DATABASE_PLATFORM
#  exit 1
#  ::
#  esac
#
#
# # Check for correct setting of DB_UNICODE_FLG
# # case $DB_UNICODE_FLG in
# # x

Figure 4-10  Dev2prod.ksh
Figure 4-11 shows the beginning of the execution of dev2prod.ksh.

```
E:\sea621\db2udb\dbserver\DB2UDB\sh dev2prod.ksh
Please review the current variable definitions for correctness:
...
SRC_USR = SADMIN
SRC_PSD = SADMIN
SRC_TBLO = SIEBEL
SRC_TBLO_PSD = SIEBEL
SNC_REPO$ = Siebel Repository
SRC_ODB = Siebel\server\siebeloracle
TGT_USR = SADMIN
TGT_PSD = SADMIN
TGT_TBLO = SIEBEL
TGT_TBLO_PSD = SIEBEL
TGT_REPO$ = Siebel Repository
TGT_ODB = Siebel\server\siebeldb2
DATA_AREA = SIEBEL_4K
INDX_AREA = SIEBEL_IDX
LONG_AREA = SIEBEL_4KL
DB16K_AREA = SIEBEL_16K
DATABASE\PLATFORM = DB2udb
DB\UNICODE\PLG = N
SIEBEL\HOME = E:\sea621\db2udb\siebelr
...
```

Ensure that the target repository does not already exist.

Check if these parameters are correct
Press Y then Enter to continue or
Press any other key then Enter to cancel

Figure 4-11  Dev2prod settings and execution

We press Y, then Enter. We get the screen in Figure 4-12, which shows the export of the source repository tables.

```
E:\sea621\db2udb\dbserver\DB2UDB\sh dev2prod.ksh
immediately at (658) 275-5800.
Connecting to the database...
Connected.
Making 2nd connection to exported datasource: Siebel\server\siebeloracle...
Connected.
Starting common api.
Process Name: Repository Import/Export Utility, Status: Started, Parameter: , Message:
Exporting Tables
Exporting table  S_APPLET ... exported  3515 rows
Exporting table  S_APPLET_META MI ... exported  413 rows
Exporting table  S_APPLET_SCRIPT ... exported  47 rows
Exporting table  S_APPLET_TOOLGLE ... exported  317 rows
Exporting table  S_APPLET_UFPROP ... exported  1366 rows
Exporting table  S_APPLICATION ... exported  22 rows
Exporting table  S_APPLET_EUD_SUSC ... exported  3 rows
Exporting table  S_APPLET_FIND ... exported  58 rows
Exporting table  S_APPLET_META MI ... exported  0 rows
Exporting table  S_APPLET_SCRIPT ... exported  0 rows
Exporting table  S_APPLET_TOOLGLE ... exported  43 rows
Exporting table  S_APPLET_UFPROP ... exported  0 rows
Exporting table  S_APPLET_UFPROP ... exported  704 rows
Exporting table  S_APPLET_UFPROP ... exported  9866 rows
Exporting table  S_ARGM_ATTR ... exported  59 rows
Exporting table  S_ARGM_ATTR ... exported  136 rows
Exporting table  S_ARGM_ATTR ... exported  65 rows
```

Figure 4-12  Export of source repository tables
At this point we have moved the repository to the target and synchronized the logical and physical schema at the target environment.

Recompiling the SRF file
Every time the repository gets updated, we need to recompile the SRF file.

From the client workstation, go to Siebel Tools->Repository, option Compile as shown in Figure 4-14 on page 65.
Figure 4-14  Compiling the SRF file - 1

The Object Compiler screen appears, as shown in Figure 4-15 on page 66.
Figure 4-15  Compiling the SRF file - 2

Select **All projects**, give a name to the Siebel repository file (for example, Siebelora), and save by default into the tools/objects directory as siebelora.srf. Click **Compile** to compile the new srf file.

After successful compilation, all clients need to pull this new srf file into their local client objects directory.

**Exporting data with Dataexp**

Dataexp will export both seed data and user data from the source tables.

It will use as input file the list of table names we have generated and saved in ora_all_out.inp file (see “Generating a list of tables to migrate” on page 56). It will import the tables listed in the input file ora_all_out.inp.

Siebel Systems provides an execute module named dataexp.exe that you will find by navigating to the e:\sea621\oracle\siebsrv\bin directory. Obtain a description of the parameters for exporting data by entering dataexp/? at the command line.

Execute this module by entering the following command:

```
e:\sea621\oracle\siebsrv\bin\dataexp /u SIEBEL /p SIEBEL /c SiebSrvr_SiebelOracle /d SIEBEL /f SIEBEL.DAT/I ora_all_out.imp
```
This job creates the output file called siebel.dat. This file will be used as input by the dataimp.exe module to import the data to DB2 for NT.

Importing data with Dataimp
Dataimp will import the output of Dataexp into the target tables. It uses as input the siebel.dat file generated by Dataexp.

Navigate to the e:\sea621\oracle\siebsrvr\bin directory to find dataimp.exe. Obtain a description of the parameters by entering dataimp/? at the command line.

Execute this module by entering the following command:
```
e:\sea621\oracle\siebsrvr\bin\dataimp /u SIEBEL /p SIEBEL /c SiebSrvr_SiebelDB2 /d SIEBEL /f siebel.dat
```

At the end of this execution you have imported the seed data and all user data to the target Siebel-DB2 database on NT.

Validating the migration
You can now go to Siebel Call Center and verify that the migrated user data is there.

4.2.2 Siebel Dataexp/Dataimp - no schema changes

This method can be used when there are no schema changes between the source and target environments (typically a first time migration). It includes the following steps:
- Generating a list of tables to migrate, and cleaning up target tables
- Running Dataexp to export all source tables (including the repository, seed data, and user data) from Oracle for NT
- Running Dataimp to import the output of the previous step into the target tables on DB2 for NT
- Validating the migration

Generating a list of tables to migrate
Before migrating the data, we need to determine which tables need to be migrated. We will migrate all tables including the repository, the seed tables, and user tables.

From the Windows desktop, go to Start->Programs->Oracle->Application Development->SQL Plus and issue the SQL statement shown in Figure 4-16 on page 68.
Figure 4-16  SQL statement to generate table names

The output of this SQL statement will give us a list of table names we will want to migrate. Consolidate the table names and save this output as ora_all_out.inp, for example. We will use this list as input to the Dataexp utility later.

Cleaning up target tables

We need to make sure there is no data in the target tables before we start migrating the data. We want to avoid running into duplicate data issues.

Go to SQL Plus and build the SQL statement shown in Figure 4-17.

Figure 4-17  SQL statement to generate SQL DELETE statements

Running this SQL statement produces output that builds the SQL DELETE statements required to clean up the target tables. Consolidate the output and save it as del_data_run.inp, for example, and execute it in DB2 Command Center before you run Dataimp. After the execution the target tables will be cleaned up.

Exporting data with Dataexp

Dataexp will export repository, seed, and user data from the source tables.
Chapter 4. Migrating Siebel data from Oracle for NT to DB2 for NT

It will use as input file the list of table names we have generated and saved in the ora_all_out.inp file (see “Generating a list of tables to migrate” on page 56). Dataexp will import the tables listed in the input file ora_all_out.inp.

Siebel Systems provides an execute module named dataexp.exe that you will find by navigating to the d:\sea621\oracle\siebsrvr\bin directory. Obtain a description of the parameters for exporting data by entering dataexp/? on the command line.

Execute this module by entering the following command:

```
e:\sea621\oracle\siebsrvr\bin\dataexp /u SIEBEL /p SIEBEL /c SiebSrvr_SiebelOracle /d SIEBEL /f e:/Mov_dat/ora_all.dat /i d:/Mov_dat/ora_all_out.inp
```

**Note:** The input file ora_all_out.inp is generated from the S_table, which contains the logical schema. It is therefore possible to have tables that are not physically created (for example, EAI_...). Dataexp complains about not physically finding those tables. Ignore these messages.

Dataexp creates an output file we called ora_all.dat. This file will be used as input by the dataimp.exe module to import the data to the DB2 for NT.

**Importing data with Dataimp**

Dataimp will import the output of Dataexp into the target tables. It will use the ora_all.dat file as input file.

Navigate to the e:\sea621\oracle\siebsrvr\bin directory to find dataimp.exe. Obtain a description of the parameters by entering dataimp/? at the command line.

Execute this module by entering the following command:

```
e:\sea621\oracle\siebsrvr\bin\dataimp /u SIEBEL /p SIEBEL /c SiebSrvr_SiebelDB2 /d SIEBEL /f e:/Mov_dat/ora_all.dat
```

At the end of this execution you have imported the seed data and all user data to the target Siebel-DB2 database on NT.

**Validating the migration**

You can now go to Siebel Call Center and verify that the migrated user data is there.
4.2.3 Siebel EIM

We tested the EIM method to migrate the Siebel database from Oracle to DB2 for NT. EIM can be used to migrate both small and large volumes of data. Figure 4-18 describes the Oracle-DB2 data migration using EIM.

![Figure 4-18 Migrating Siebel data from Oracle to DB2 using EIM](image)

This EIM process is very similar to the one explained in Chapter 3, "Migrating Siebel data from DB2 for NT to DB2 for OS/390" on page 31.

We perform the following processes:

- Using Siebel Tools, rename the repository at the target.
- Move the repository and synchronize the schema with Dev2prod.
- Compile the new SRF file.
- Identify the EIM staging tables that map the base tables.
- Run EIM Export to get the data into the EIM staging tables.
- Run Dataexp to unload EIM staging tables into a flat file.
- Run Dataimp, which imports the content of the flat file into the EIM staging tables at the target.
- Run EIM Import to move the EIM staging tables into the base tables at the target.
Renaming the repository
Using Siebel Tools from the client workstation, rename the repository as explained in “Renaming the repository” on page 60.

Moving the repository and synchronizing the schema
We use the Dev2prod utility to migrate the repository and synchronize the schema. Dev2prod compares the source and target schema and implements only the differences at the target. Refer to “Moving the repository and synchronizing the schema” on page 61 to see how to run this process.

Recompiling the SRF file
Recompile the SRF file as described in “Recompiling the SRF file” on page 64.

Running EIM Export
First, make sure the EIM source tables are clean. Refer to “Cleaning up target tables” on page 57 to see an example of how to clean the tables. 3.4, “Siebel EIM” on page 38 gives detailed explanations on how to run EIM Export. The process with Oracle is exactly the same.

Running Dataexp and Dataimp
Run Dataexp and Dataimp as explained in 3.4.2, “Dataexp” on page 43 and 3.4.3, “Dataimp” on page 44.

Running EIM Import
Run EIM Import as explained in 3.4.4, “EIM Import” on page 45. Make sure the EIM target tables are clean.
This chapter investigates the methods for migrating the Siebel database from a source Oracle for NT environment to a target DB2 for OS/390 environment.

We investigated the following alternatives:

- **Two-phase migration**
  - Oracle for NT => DB2 for NT => DB2 for OS/390

- **One-phase migration**
  - Oracle for NT => DB2 for OS/390
  
Using three methods:
- Dataexp/Dataimp
- Dataexp/FTP/DB2 Load
- Relational Connect/DB2 Load
5.1 Two-phase migration - Oracle for NT to DB2 for OS/390

Figure 5-1 shows the two-phase migration from Oracle for NT to DB2 for OS/390.

The two-phase migration consists of the following:

- Phase one: migrate Oracle for NT to DB2 for NT
  This migration process is described in Chapter 4, “Migrating Siebel data from Oracle for NT to DB2 for NT” on page 53.

- Phase two: migrate DB2 for NT to DB2 for OS/390
  This migration process is described in Chapter 3, “Migrating Siebel data from DB2 for NT to DB2 for OS/390” on page 31.

The advantage of the two-phase migration is that it breaks the migration process into smaller steps so that different skill sets can accomplish each step.

- The first phase, Oracle for NT to DB2 for NT, requires mostly a DBA familiar with the NT platform and with Oracle and DB2 skills. This phase also makes it possible to detect and resolve some of the migration issues on the NT tier, taking advantage of the DBA skills available on this platform.

- The second phase, DB2 for NT to DB2 for OS/390, requires mostly DB2 for OS/390 DBA skills with some understanding of DB2 for NT and its connectivity to the DB2 for OS/390 platform.

Each phase can be planned at different times with different people. Little interaction is required between the different skill sets.
The downside of this method is that it takes more time because the data is moved twice, but if the volume of data to move is small, the time it takes to move the data might not be an issue. If the volume of data to move is large, the time it takes to move the data might then become an issue.

### 5.2 One-phase migration - Oracle for NT to DB2 for OS/390

Figure 5-2 shows the one-phase migration from Oracle for NT to DB2 for OS/390.

![Figure 5-2 One-phase migration: Oracle for NT to DB2 for OS/390](image)

The advantage of the one-phase migration is that it is faster than the two-phase migration. You move the data once. With large volumes of data to move, this method is better.

On the other hand, this method requires a higher interaction of skill sets, such as Oracle/DB2 for NT and DB2 for OS/390.

We consider the following methods here:

- Dataexp/Dataimp
- Dataexp/FTP/DB2 Load
- Relational Connect/DB2 Load
5.3 Dataexp/Dataimp - no schema changes

This migration scenario assumes that there are no source object names that violate the DB2 for OS/390 naming conventions. There are no schema changes to take care of at the target environment.

The migration process, with no schema changes, is straightforward and very similar to the one described in Chapter 3, “Migrating Siebel data from DB2 for NT to DB2 for OS/390” on page 31. The migration process includes the following steps:

- Generate a list of source tables to migrate
- Clean up target tables on DB2 for OS/390
- Export data from Oracle for NT with Dataexp
- Import data into DB2 for OS/390 with Dataimp
- Validate the migration

5.4 Dataexp/Dataimp - with schema changes

You will typically have schema changes if you have customized, in your source Oracle environment, any database objects that violate the size limitations imposed by the target DB2 for OS/390 environment. You will have to fix those size limitation issues when migrating to the DB2 for OS/390 environment.

Siebel Systems and the IBM Hursley laboratory tested this scenario in a joint effort as a proof of concept at a customer site in UK.

5.4.1 Verifying object names

DB2 for OS/390 allows a maximum of 18 characters for table, index and column names, and up to 255 bytes for index key length. If you have any source table, column and index names that exceed these limitations, you must address the issues in one of the following ways:

- Consolidate object names in the source environment before starting the migration. This is the recommended approach because it brings both the source and target environments in line with the DB2 for OS/390 naming conventions, which will be used once the migration is accomplished.
- Consolidate object names during the migration. This is doable but is not the simplest and easiest way of accomplishing it.
5.4.2 Consolidating object names before the migration

The migration process is as follows:

- Run the validate_obj_names.ksh script against the source Oracle database. Note that this script is shipped with the DB2 for OS/390 installation. Navigate to the d:\sea621\dbsrvr\db2390 subdirectory to find it. If you do not have DB2 for OS/390 install scripts available on your source environment, you will not see this script. In this case you can still directly execute the relevant SQL statements (see Appendix F, “Validating object names” on page 169).

The validate_obj_names.ksh script generates the List of Invalid Objects Report, which lists all objects that exceed the DB2 for OS/390 size limitations for table, column and index names and sizes.

Review the report and fix all source objects that violate the maximum length limit for DB2 for OS/390. Do this on the source Oracle environment before starting the migration.

Make sure that the DB2 for OS/390 tables are also customized accordingly at the target site. The DBA can do the customization manually on OS/390.

- Generate a list of tables to migrate using the following SQL at the source site:

  ```sql
  SELECT NAME FROM SIEBEL.S_TABLE
  WHERE (TYPE LIKE 'DATA%' OR TYPE = 'EXTENSION')
  AND NAME NOT IN ('S_DOCK_STATUS','S_DOCK_TXN_LOG')
  ORDER BY NAME
  ```

  The output of the SQL statement gives the list of source tables to migrate. Save them into a file that will be used as input to Dataexp.

- To clean up target tables, use the following SQL at the target site to generate the SQL DELETE statements:

  ```sql
  SELECT 'DELETE FROM SIEBEL.' || NAME || '; FROM SIEBEL.S_TABLE
  WHERE (TYPE LIKE 'DATA%' OR TYPE = 'EXTENSION')
  AND NAME NOT IN ('S_DOCK_STATUS','S_DOCK_TXN_LOG')
  ORDER BY NAME
  ```

  The output of this SQL statement builds the SQL DELETE statements. Save those statements into a file. Run those DELETE statements against DB2 for OS/390.

- Rename the target repository using Siebel Tools. See “Renaming the repository” on page 60.

- Move the repository and synchronize the schema using dev2prod. See “Moving the repository and synchronizing the schema” on page 61.

- Recompile the .SRF file using Siebel Tools. See “Recompiling the SRF file” on page 64.
5.4.3 Consolidating object names during the migration

Alternatively, you can consolidate object names during the migration process.

Figure 5-3 shows an Oracle for NT to DB2 for OS/390 migration using the Dataexp/Dataimp process, including object name size modifications.

The migration process is as follows:

- Generate a list of tables to migrate using the following SQL:

  ```sql
  SELECT NAME FROM SIEBEL.S_TABLE
  WHERE (TYPE LIKE 'DATA%' OR TYPE = 'EXTENSION')
  AND NAME NOT IN ('S_DOCK_STATUS','S_DOCK_TXN_LOG')
  ORDER BY NAME
  
  The output of the SQL statement gives the list of tables to migrate. Save it into a file.

- Clean up target tables in DB2 for OS/390 using the following SQL to generate the SQL DELETE statements:

  ```sql
  SELECT 'DELETE FROM SIEBEL.' || NAME || ';' FROM SIEBEL.S_TABLE
  WHERE (TYPE LIKE 'DATA%' OR TYPE = 'EXTENSION')
  ```
AND NAME NOT IN ('S_DOCK_STATUS','S_DOCK_TXN_LOG')
ORDER BY NAME

The output of this SQL statement builds the SQL DELETE statements. Save them into a file. Run these DELETE statements against DB2 for OS/390.

Export data with Dataexp using the text format option. The last switch, /Tt, makes it possible to have the file in text format.

dataexp /U sadmin /P sadmin /C SiebSrvr_SiebelOracle /D siebel /F dataexp.dat /I dataexp.inp /l dataexp.log /Tt

This lets you have not only the data but the table and column names in the same file. You can then edit the .dat file.

- Edit the .dat file
  - First, determine which source objects need to be modified to conform to DB2 for OS/390 size limitations. Use the validate_obj_names.ksh script, which generates the List of Invalid Objects, or run the SQL statements described in Appendix F, “Validating object names” on page 169, to determine the offending object names and index key sizes.
  - Once you have the list of the offending object names, you can update the .dat file, which is in text format. “The .dat file” on page 171 shows a one-table example of a .dat file in text format.

- Make sure that the corresponding DB2 for OS/390 objects are correctly defined and created at the target site.

- Rename the repository at the target using Siebel Tools. See “Renaming the repository” on page 60.

- Move the repository and synchronize the schema with dev2prod. See “Moving the repository and synchronizing the schema” on page 61.

- Import data with Dataimp

  To speed up the import process into DB2 for OS/390, you can also split Dataimp into multiple batch processes by generating multiple dataexp .dat files. For example, the SQL that extracts the table names can be split by type of tables or alphabetic order. Once this is done, several Dataimps can execute in parallel against each .dat file previously generated.

- Use Siebel Tools to update the business components in the repository at the target OS/390 site. Business components that reference the table, index, and column names we changed must also be updated. This extra step is needed to make sure that the business logic is using the correct object names.
  - The business components need to call the right column and field names.
  - Scripts, if any, also need to be changed.

  From Siebel Tools, you can also export the business component as an .SIF file, edit and change it, and import the modified file into Siebel Tools.
Next, the .SRF file needs to be created by recompiling in Siebel Tools. This .SRF needs to be distributed to all clients.

Finally, you validate the migration by connecting to Siebel Call Center and verifying that the migrated data is there.

5.5 Dataexp/FTP/DB2 Load

The Dataexp/FTP/DB2 Load migration method is an alternative to Dataexp/Dataimp. When large volumes of data are involved, Dataimp can become inappropriate for moving the data over to the OS/390 platform for the following reasons:

- Dataimp uses dynamic SQL INSERTs to load the target DB2 for OS/390 tables remotely over a DRDA connection. A local batch process, such as the DB2 Load utility on OS/390 can perform faster in loading large volumes of data.
- With large volumes of data to load, the dynamic SQL technique causes a flooding of the logs and generates many archive logs on the OS/390 site.
- Dataimp does not issue COMMITs. As a result, parts of the DB2 for OS/390 catalog are locked out, preventing any other dynamic SQL programs from running.
- Many page splits may also occur in the indexes.

Migration process

We recommend that you consolidate object names up front in the Oracle source environment before starting the migration. The first part of this process is the same as the one described in 5.4.2, "Consolidating object names before the migration" on page 77.

The migration process is as follows:

- Consolidate object names. Run the validate_obj_names.ksh script against the source Oracle database. Review the List of Invalid Objects Report, which lists all objects that exceed the DB2 for OS/390 size limitations for table, column and index names and sizes. Fix the problem in the Oracle source environment before starting the migration. Consolidate object names and sizes accordingly on the target DB2 for OS/390 site.

- Generate a list of source tables to migrate.
- Clean up target tables.
- Rename the target repository using Siebel Tools.
- Move the repository and synchronize the schema at the target.
Recompile the SRF file using Siebel Tools.

Export data (in binary format) from Oracle using Dataexp:

```
e:\sea621\oracle\siebsrvr\bin\dataexp /u userid /p password /c ODBC name/d tablecreator /f outputfilename /t binary /i tablenamesfile
```

At the end of Dataexp execution, a report is produced stating how many rows have been extracted from each table. Note this information to check and validate your migration later.

Dataexp produces a .dat output file. FTP this file to OS/390 using the binary option to ensure that it is sent as an ASCII file.

Size the DB2 for OS/390 table spaces. You can do this manually or automate it as follows:

- Produce, at the source site, a list containing the table name and a count of the number of rows in that table.
- FTP that file to OS/390. It will be read in by the REXX procedure.
- Develop a REXX procedure to generate, for each table space, an ALTER statement with a PRIQTY based on the maximum length of records in each of the table spaces. An example of such a REXX procedure is provided in “REXX procedure” on page 175.
- Execute the REXX procedure using “JCL to run the REXX procedure” on page 178. The REXX procedure builds the ALTER statements and saves them into a file. The following are examples of such ALTER statements:

```
ALTER TABLESPACE SIEBEL00.DDDDD601
PRIQTY 684 SECQTY 68;
ALTER TABLESPACE SIEBEL00.DDDDD602
PRIQTY 56 SECQTY 5;
ALTER TABLESPACE SIEBEL00.DDDDD603
PRIQTY 1 SECQTY 1;
ALTER TABLESPACE SIEBEL00.DDDDD604
PRIQTY 1 SECQTY 1;
ALTER TABLESPACE SIEBEL00.DDDDD606
PRIQTY 1 SECQTY 1;
ALTER TABLESPACE SIEBEL00.DDDDD607
PRIQTY 1 SECQTY 1;
```

- Execute the ALTER statements at the target DB2 for OS/390 site.

At this point, you need to develop a program to transform the format of Dataexp output (.dat file) into DB2 internal format. An example of such a program is provided in , “PLFSIEBL COBOL program” on page 179.
You must compile this program on OS/390 with the TRUNC(BIN) option. COBOL, by default, truncates on the number in the picture clause. So PIC S9(4) COMP, for example, would hold a number up to 9999. But DB2 SMALLINT can hold a number up to 32767. If TRUNC(STD) is used, then 32767 in DB2 would be seen as 2767 by COBOL. TRUNC(BIN) truncates at the byte level, so COBOL will still see 32767.

The program also handles the date format differences between Oracle (10 characters) and DB2 (26 characters). It pads the Oracle date fields with zeros on the right up to 26 characters.

PLFSIEBL is a COBOL program that extracts the data from the .dat file and creates load data in DB2 internal format. It also generates the DB2 Load utility cards. A LOADFILE and a LOADCARDS file are created for each table space as shown in Figure 5-4.

Run the PLFSIEBL program using the “JCL to run the PLFSIEBL program” on page 204.

After the LOADFILE and LOADCARDS files are created, the next step is to run them through the DB2 LOAD utility with the LOG=NO option. See “JCL to run the DB2 LOAD utility” on page 205. Table spaces can be loaded in parallel by running multiple Load jobs. Statistics are updated as part of the Load job.
Validate the migration. You should check that the number of rows loaded at
the target corresponds to the number of rows extracted by Dataexp and put in
the .dat file.

The final step is to take an image copy.

5.6 Relational Connect/DB2 LOAD

This data migration tool was designed and prototyped by the DB2 for OS/390
Porting Center at IBM Silicon Valley Laboratory (SVL). It extends and refines an
approach used successfully for migrating Siebel database data from DB2 for NT
to DB2 for OS/390, and was architected to meet the following objectives:

- Perform well when migrating gigabytes or terabytes of data
- Minimize the use of file I/O and DASD resources
- Minimize datatype conversions
- Reduce the complexity of migrating data from Oracle to DB2 for OS/390

Other approaches for migrating large quantities of data from relational data
stores to DB2 for OS/390 often do not perform well and are overly complex. Poor
performance and complexity can be attributed to a variety of factors, including
the use of SQL to both extract the data from the source data store and place the
data into DB2 for OS/390; the writing of a large number of records to the DB2 log,
the storing of the data in one or more intermediate data stores (e.g., local file
systems); performing multiple datatype conversions as the data is moved
between environments; the use of less efficient data transfer mechanisms; and
the use of processes that are not fully automated.

This data migration tool uses DB2 for OS/390 bulk data loading interfaces (i.e.,
the DB2 LOAD Utility) to place the data into DB2 for OS/390. LOAD LOG(NO) is
specified for the load so that no logging occurs during the load process. The jobs
that extract the data from the source data store and place the data into DB2 for
OS/390 execute in parallel. The tool eliminates the use of intermediate data
stores, minimizes the datatype conversions that are performed, and transfers the
data from the source machine to the target machine in large blocks, using the
efficiencies of the DRDA protocol. In addition, the setup, creation, and execution
of the data migration jobs is fully automated.

Migration process

Figure 5-5 on page 84 illustrates the migration process using the Relational
Connect/DB2 Load Utility method.
The data migration process is initiated by DMTFUNLD, a batch application that executes under OS/390. Each invocation of the application does the following:

1. Unloads a table from an Oracle database using Relational Connect and DB2 for NT by executing a `SELECT * FROM tablename` statement.
2. Uploads the table to the OS/390 machine using DRDA flows.
3. Generates the DB2 for OS/390 LOAD Utility control statement for the subsequent load of the table into DB2 for OS/390.

The LOAD Utility control statement has the following form:

```
LOAD DATA LOG NO INDDN ddname ENFORCE NO RESUME YES
SORTKEYS 150000000 INTO TABLE into_table_specification
```

The batch application is designed to pass the LOAD Utility control statement and the data to the DB2 for OS/390 LOAD Utility using Batch Pipes.

Batch Pipes are a performance option that can significantly cut down the overall elapsed time of the migration process. They allow you to:

- Parallelize the unload of the source data and the load into the target database
- Reduce the number of I/O operations and the use of DASD by transferring data through processor storage rather than to and from DASD
Figure 5-6 explains the Batch Pipes concept.

With Batch Pipes you can pipe the output of DMTFUNLD directly into the DB2 LOAD Utility input as shown in Figure 5-7.

The use of Batch Pipes is recommended but not required.

In addition, multiple instances of DMTFUNLD can be executed concurrently so that multiple tables can be migrated in parallel from the Oracle server to DB2 for OS/390.
The tool also has the capability to create a copy of the data being loaded—in parallel with the execution of the extraction and the load. This copy is useful in situations where the amount of data being migrated is large and the LOAD Utility fails to execute to completion because it permits a restart from the last internal LOAD Utility checkpoint rather than from the start of the load phase. This optional facility does create a copy of the data in an intermediate data store, the local OS/390 file system.

This discussion of the migration tool makes the following assumptions about the Oracle and DB2 schemas:

- The schemas contain the same tables (for example, all of the Siebel database objects and data will be migrated from the Oracle source system to the DB2 for OS/390 target system).
- The DB2 tables will have the same table and column names as the Oracle tables. The Oracle datatypes associated with columns will be replaced with equivalent DB2 datatypes.

These assumptions permit the existing Siebel applications to access the DB2 for OS/390 tables and preserve the referential integrity relationships that are introduced and managed by those applications. If the Oracle and DB2 schemas are not the same, then the source and target schemas need to be synchronized prior to the use of the migration tool. This can be accomplished using tools described in the Siebel Tools Reference, which is contained in the Siebel Bookshelf.

The steps to migrate are as follows:

- Install the source code for DMTFUNLD, the JCL for the precompile, assemble, link-edit, and bind of DMTFUNLD, and the DMT REXX EXEC in your favorite OS/390 assembler source code, JCL, and REXX libraries.
- Install the DMTASU REXX EXEC in your favorite NT REXX directory.
- Precompile, assemble, link-edit, and bind the plan for DMTFUNLD. Grant EXECUTE authority on the plan for DMTFUNLD to the user or user IDs that will be performing the data migration.
- Configure DB2 for NT V7 and DB2 Relational Connect as a federated system for access to Oracle data.
- Modify the configuration of DB2 for NT so that DB2 can retrieve data from large Oracle tables containing one or more columns having a datatypem of long.
- Generate a list of source tables to be migrated using the REXX EXEC DMTASU.
- Migrate the Siebel database schema from Oracle to DB2 for OS/390.
Configure the DB2 for OS/390 Communications Database so that DB2 for OS/390 can function as a DRDA Application Requester to access the remote Oracle server via DB2 for NT.

Use the REXX EXEC DMT to generate the JCL for the unload and load of the source tables to be migrated. The JCL generated for the unload initiates and synchronizes both the unload and load jobs.

Clean up the target tables.

Execute the JCL for the unload.

Validate the migration.

Performance results
The elapsed time for migrating a table depends on a number of factors, including the size of the table, the number of columns, the number of columns of type Oracle long, and the number of indexes created on the table.

A performance assessment of this data migration tool is currently underway. In one of our early performance runs, we migrated the table S_NOTE_OPTY in 1 hour and 13 minutes. S_NOTE_OPTY contains 12 columns, has one column of type Oracle long, and has two indexes: S_NOTE_OPTY_F2 and S_NOTE_OPTY_P1. Our instance of this table contained 2,056,254 rows or about 1.54 gigabytes of data.

Availability of the Relational Connect/DB2 Load Utility tool
At the time of writing, the Relational Connect/DB2 Load Utility tool is not generally available for use by customers. The use of this tool can be made available in customer environments through DB2 for OS/390 Data Management Consulting Services at IBM Silicon Valley Laboratory. Please contact your IBM representative for further details.
Implementing the Siebel source environment on NT

This section is intended for those who want to create a test migration environment and need to set up a source NT environment from which to migrate.

Those who already have a source environment, such as a production environment, can skip this section and go directly to Chapter 2, “Implementing the target Siebel environment on OS/390” on page 11.
Setting up a source environment

In order to document our migration scenario, we needed a source Siebel system to migrate from. Figure A-1 shows the environment we created for this purpose.

![Diagram of Siebel source environment on NT]

To achieve our Siebel implementation on NT we followed the guidelines described in *Installing Siebel 2000 On Microsoft Windows NT 4.0 Using IBM’s DB2 UDB 6.1*, SWG CECI Bulletin, which you can find only on the following IBM internal Web site:


We found this document very useful; it saved us a lot of time. It is reproduced in this section, updated with the values we used for our implementation.

This section contains the procedures we used to create our Siebel Server environment on NT, supporting the Siebel database on DB2 for NT. The scenario does not intend to provide a production environment, but was designed rather to let us install just enough to allow us to bring up a Siebel 2000 system to use as a source environment to migrate from.

We discuss the following:
- Installing the base NT software
- Installing DB2 for NT
- Configuring DB2 for Siebel 2000
- Creating the Initial DB2 database
- Installing the Siebel Gateway Server
- Installing the Siebel Server
- Installing the Siebel Database Server
- Initializing the Siebel database
- Verifying the install
Installing the base NT software

The following product set is required for installing the Siebel application on NT:
- Microsoft Windows NT 4.0 Server with Service Pack 5 or 6
- Microsoft Office 2000
- Adobe Acrobat Reader 4.0
- Microsoft Internet Explorer 5.0
- IBM DB2 EE for NT V6.1 with DB2 Fix Pack 6s (We used Fix Pack 3sa because it was the only one available when we did our tests).
- Siebel 2000 V6.2.1 for Microsoft Window

The implementation procedures require a number of files to be downloaded from the Web. We have listed them here in case you want to download them before they are called out in the procedures:
- DB2 Connect EE Fix Pack:
- Microsoft Windows NT Service Pack 5:
- We already had Adobe Acrobat Reader on our machine, but if you do not, it is available at:

Installing Microsoft Windows NT 4.0 server

We chose NT as our application server platform and installed Microsoft Windows NT 4.0 server with Service Pack 6. The installation procedure is as follows:

1. Install Microsoft Windows NT 4.0 server with Service Pack 6.

   Our recommendation for hardware includes the following:
   - Minimum processor speed of 300 MHz
   - 512 Mb of RAM
   - 10 GB of DASD, partitioned into a minimum of two logical drives:
     a C drive of 2 GB, and a D drive of the remaining 8 GB, both formatted for the NTFS file system
2. Create an appropriate machine name (ITSONT0, in our example).
3. Make sure that the TCP/IP protocol is operational.
4. Make sure that Internet Explorer is configured and working.

**Installing Microsoft Office 2000**

Siebel requires the Microsoft Office suite for their solution; even though we did not use it for our project, the Siebel installation procedures require it.

1. Insert the Microsoft Office 2000 CD-ROM into the drive and wait for the Installation setup to start. If it does not start automatically, select Start->Programs->Windows NT Explorer and double-click the Setup.exe program in the root directory of the CD-ROM to start the installation.

2. Enter your Customer Information and click Next >.

3. Select I accept the terms in the license Agreement and click Next >.

4. Click Customize..., choose the location where you want Microsoft Word 2000 to be installed, and click Next >.

5. Make sure that the Update Windows to Include: screen shows Microsoft Internet Explorer 5.0 - Standard in the pick list, and click Next >.

6. Press the v beside Microsoft Word for Windows and select Run all from My Computer. Repeat for Converters and Filters.

7. Click Install Now.

8. When prompted regarding whether or not you want to restart your system, click Yes to restart.

9. After Windows Update has completed, the message Finishing Microsoft Office 2000 Setup will appear.

10. Verify that Microsoft Word is available by selecting Start->Programs->Microsoft Word and seeing that Word opens.

**Installing Adobe Acrobat**

The Siebel application requires the Adobe Acrobat reader (note that the requirement is Adobe Acrobat 3.x or later).

We already had Adobe Acrobat reader 4.0 on our system.

**Installing Internet Explorer**

When we installed Microsoft Office 2000, it automatically upgraded to Microsoft Internet Explorer 5.0 (this is a function of installing Office 2000).
Installing DB2 for NT

This section describes the implementation of our DB2 for NT environment, including the creation of a DB2 Startup Account, installation of the DB2 server, and verification of the installation.

Creating the DB2 startup account

To create the DB2 user startup account, do the following:

1. Select Start->Programs->Administrative Tools(Common)->User Manager for Domains.
2. Make SADMIN a member of the Administrators group.
3. Make SADMIN the password and set the password to never expire.
4. Add Act as Part of the Operating System Rights in NT.
5. In User Manager for Domains, select Policies->User Rights...
   Click the box to the left of Show Advanced User Rights.
6. From the Rights pull-down, select Act as Part of Operating System.
7. If the SADMIN account does not appear in the Grant To box, click Add..., click Show Users, scroll down in the Names box and select sadmin, and press Add followed by OK to add the account.
8. Follow the same procedure with Logon as a service.
9. Click OK.

Installing the DB2 server

To install DB2 for NT V6, do the following:

1. Log off and log in to the system, using the newly created sadmin account.
2. Shut down any other running programs so that the setup program can update files as required.
3. Insert the DB2 for NT CD-ROM into the drive and wait for the Installation setup to start. If it does not start automatically, select Start->Run..., click Browse..., navigate to the \EE subdirectory, double-click the Setup.exe program, and click OK to begin the installation.
4. When the Welcome to IBM DB2 Universal Database for Windows 32-bit operating systems screen appears, click Next>.
5. When the Select Products screen appears, make sure that the DB2 Enterprise Edition check box is checked and click Next >.
6. Click Typical Install.
7. Set the destination folder and click **Yes** to create the folder (if it doesn’t already exist).

8. Click **Next >**.

9. On the Enter Username and Password for the Administration Server screen, enter `sadmin` for the Username, Password, and Confirm password fields and click **Next >**.

10. You may receive the message Setup is unable to validate the username `sadmin`...; click **OK** to continue.

11. When the Start Copying Files screen appears, click **Next >** to begin the file copying process.

12. When the Setup Complete screen appears, select **Yes, I want to restart my computer now** and click **Finish**.

13. Install the DB2 run-time client.

### Installing the Siebel 2000 DB2 for NT Fix Pack

Currently the Fix Pack level available for Siebel 2000 is Fix Pack 6s. This is the one you should use. We used Fix Pack 3sa because it was the only one available at the time we did our tests. To install the Fix Pack, do the following:

1. Download the Siebel2000/DB2 Fix Pack from:
   

2. Log in to the system using the `sadmin` account.

3. Select **Start->Programs->Windows NT Explorer**, navigate to the directory in which you downloaded the Siebel 2000/DB2 Fix Pack, double-click **Setup.exe**, and click **OK** to begin the installation.

4. When the WARNING screen appears, indicating that there are DB2 processes currently running and locked, click **YES** to stop the processes.

5. When the Welcome screen appears, click **Next >**.

6. When the Choose Destination Location screen appears, click **Next >**.

7. When the Start Copying Files screen appears, click **Next >**.

8. When the Setup Complete screen appears, select **Yes, I want to restart my computer now** and click **Finish**.

### Verifying the DB2 installation

Verify the installation as follows:
1. Start First Steps by choosing **Start->Programs->DB2 for Windows NT->First Steps**.

2. Click **Create the SAMPLE Database** on the main panel.

3. When the DB2 First Step screen appears, click **Yes** to continue.

4. The command may take a few minutes. When the SAMPLE database has been created successfully, you receive a completion message.

5. Click **OK**.

6. Once the database has been created, click the **View the SAMPLE Database** to select data from the SAMPLE database. Specify **sadmin** when prompted for the User ID and Password fields, and press **OK** to continue.

7. Select **Script->Execute** and the results should be displayed.

8. Click the **Script** tab and select **Script->Shut Down DB2 Tools** to exit the **Command Center**.


10. After you have verified the installation, you can remove the SAMPLE database to free up disk space. Select **Start->Programs->DB2 for Windows NT->Command Window** and enter the `db2 drop database sample` command to delete the SAMPLE database.

11. Type **exit** and press **Enter** to close the Command Window.

### Configuring DB2 for Siebel 2000

In these sections we describe how to configure and customize the DB2 for NT product just installed.

We used the siebelbf.zip file we downloaded from the IBM intranet site:


Unzip siebelbf.zip to get the following files:

- `db2_db_mgr_cfg.bat`
- `db2_db2set.bat`
- `db2_dd_cfg.bat`
- `tablespace.bat`
- `forceit.bat`
- `bufpool.bat`

You can view the content of those files in Appendix B, “The .bat files” on page 117.
DB2 Database Manager configuration parameters

We set up the DB2 Database Manager configuration parameters by executing the `db2_db_mgr_cfg.bat` file in a DB2 Command Window.

Table A-1 reflects the final parameter values after the batch file completes execution.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Explanation/Comment</th>
<th>Setting/Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>UDF_MEM_SZ</td>
<td>UDF shared memory set size (1 KB)</td>
<td>1024</td>
</tr>
<tr>
<td>SHEAPTHRES</td>
<td>Sort heap threshold (4 KB)</td>
<td>100000</td>
</tr>
<tr>
<td>DIR_CACHE</td>
<td>Directory cache support</td>
<td>YES</td>
</tr>
<tr>
<td>ASLHEAPSZ</td>
<td>Application support layer heap size</td>
<td>15</td>
</tr>
<tr>
<td>RQRIOBLK</td>
<td>Maximum requester I/O block size (bytes)</td>
<td>65535</td>
</tr>
<tr>
<td>QUERY_HEAP_SZ</td>
<td>Query heap size (4 KB)</td>
<td>8192</td>
</tr>
<tr>
<td>MAXAGENTS</td>
<td>Maximum number of existing agents</td>
<td>1000</td>
</tr>
<tr>
<td>NUM_INITAGENTS</td>
<td>Initial number of agents in pool</td>
<td>0</td>
</tr>
<tr>
<td>MAX_COORDAGENTS</td>
<td>Maximum number of coordinating agents</td>
<td>MAXAGENTS$^a$</td>
</tr>
<tr>
<td>INDEXREC</td>
<td>Index re-creation time</td>
<td>RESTART</td>
</tr>
<tr>
<td>MAX_QUERYDEGREE</td>
<td>Maximum query degree of parallelism</td>
<td>1</td>
</tr>
<tr>
<td>INTRA_PARALLEL</td>
<td>Enable intra-partition parallelism</td>
<td>NO</td>
</tr>
<tr>
<td>FCM_NUM_BUFFERS</td>
<td>Number of internal communication buffers (4 KB)</td>
<td>4096</td>
</tr>
<tr>
<td>FCM_NUM_RQB</td>
<td>Number of FCM request blocks</td>
<td>1024</td>
</tr>
<tr>
<td>FCM_NUM_CONN</td>
<td>Number of FCM connection entries</td>
<td>(FCN_NUM_RQB * 0.75)$^1$</td>
</tr>
<tr>
<td>FCM_NUM_ANCHORS</td>
<td>Number of FCM message anchors</td>
<td>(FCN_NUM_RQB * 0.75)$^1$</td>
</tr>
</tbody>
</table>

$^a$ To set this value, you must input "-1" for the Setting.
Setting the DB2 environment parameters

Next, set the DB2 environment parameters. We executed the batch file db2_db2set.bat in a DB2 command window.

Table A-2 reflects the final parameter values after the batch file completes execution.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Explanation</th>
<th>Setting/Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2_HASH_JOIN</td>
<td>Turns off hash joins in Optimizer.</td>
<td>NO</td>
</tr>
<tr>
<td>DB2_RR_TO_RS</td>
<td>Improves DB2 performance with the Siebel application. Set to YES only in production environment servers.¹</td>
<td>NO</td>
</tr>
<tr>
<td>DB2_MMAP_WRITE</td>
<td>Recommended setting only; you should evaluate this setting for your particular configuration and environment.</td>
<td>OFF</td>
</tr>
<tr>
<td>DB2_MMAP_READ</td>
<td>Recommended setting only; you should evaluate this setting for your particular configuration and environment.</td>
<td>OFF</td>
</tr>
<tr>
<td>DB2_CORRELATED_PREDICATES</td>
<td>When set to YES, the optimizer is able to determine whether predicates in a query are related, which permits DB2 to calculate the filter factor more accurately.</td>
<td>YES</td>
</tr>
</tbody>
</table>

¹. The tools check-out procedure requires an isolation level of “repeatable read.” Turning this parameter on disables all repeatable reads, causing an application to use “read stability.” This status is unacceptable for tools check-out and, therefore, development purposes.

Creating the initial DB2 database

Next, create the skeleton Siebel-DB2 database with initial buffer pool and table spaces, verify that they are created correctly, configure the database, create the database user accounts, and create the shared Siebel File System.
Creating the skeleton Siebel database

To create the skeleton Siebel-DB2 database, we do the following:

1. Select Start->Programs->DB2 for Windows NT->Control Center, enter SADMIN for the User ID and Password when prompted and click OK.
2. Click the plus sign (+) beside ITSONT0.
3. Click the + beside the Instances.
4. Click the + beside the DB2 Instance.
5. Right-click Databases, then select Create Database using Smart Guide and insert the values as follows:

<table>
<thead>
<tr>
<th>New database name</th>
<th>siebeldb</th>
</tr>
</thead>
<tbody>
<tr>
<td>Comment</td>
<td>Siebel Database</td>
</tr>
<tr>
<td>Database alias</td>
<td>siebeldb</td>
</tr>
</tbody>
</table>
6. Click Done.

Creating the initial buffer pools

To create the initial buffer pools, do the following:

1. Click the plus sign (+) beside SIEBELDB.
2. Click the Buffer Pools folder to display the existing list of Buffer Pools, and in the right-hand pane, right-click the IBMDEFAULTBP buffer pool and select Alter....
3. Change the Size in 4 KB pages to 50% of available memory. Right-click the toolbar at the bottom of the screen and select Task Manager.... Then, select the Performance tab and look at Available under the Physical Memory (K) section on the right side. Divide this value by 8 to obtain the size; for example, 720,000 kilobytes of available memory / 8 or 90,000 4 KB pages.
4. At this point, we created the remaining buffer pools by executing the bufpool.bat file in a DB2 Command Window.

Verifying the buffer pools

To verify that the buffer pools were created correctly, do the following:

1. Go to DB2 Control Center ->View ->Refresh.
2. Click the plus sign (+) beside ITSONT0.
3. Click the + beside the Instances.
4. Click the + beside the DB2 Instance.
5. Click the + beside Databases.
6. Click the + beside SIEBELDB and wait for a few moments for the connect to take place.
7. Click the Buffer Pools folder and the new buffer pools should show up in the list, along with the revised setting of the IBMDEFAULTBP buffer pool.

Note that the newly created buffer pools would not become usable until the next time the database is started. You need to do this before creating new table spaces (in the following step) and associating them to the new bufferpools.

Creating the regular and temporary table spaces

Create the required regular and temporary DB2 table spaces by executing the tblspace.bat file in a DB2 Command Window.

Verifying the table spaces

To verify the table space creation, do the following:
1. Go to DB2 Control Center -> View -> Refresh
2. Click the plus sign (+) beside ITSONT0.
3. Click the + beside the Instances.
4. Click the + beside the DB2 Instance.
5. Click the + beside Databases.
6. Click the + beside SIEBELDB and wait a few moments for the connect to take place.
7. Click the Table Spaces folder and the new table spaces should show up in the list.
8. Click Control Center->Exit.

Configuring the database

Before we executed db2_db_cfg.bat to configure the database, we edited it to change the SORTHEAP parameter to a value of 32 (instead of the existing 20000) as per a Siebel consultant's recommendation. We then executed db2_db_cfg.bat in a DB2 Command Window.

The final values are reflected in Table A-3 on page 100.
### Table A-3  DB2 Database Configuration parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Explanation</th>
<th>Setting/Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>DFT_DEGREE</td>
<td>Degree of parallelism (1 - turn query parallelism off)</td>
<td>1</td>
</tr>
<tr>
<td>DFT_QUERYOPT</td>
<td>Default query optimization class</td>
<td>5</td>
</tr>
<tr>
<td>DBHEAP</td>
<td>Database heap (4 KB)</td>
<td>7429</td>
</tr>
<tr>
<td>CATALOGCACHE_SZ</td>
<td>Catalog cache size (4 KB)</td>
<td>5558</td>
</tr>
<tr>
<td>LOGBUFSZ</td>
<td>Log buffer size (4 KB)</td>
<td>512</td>
</tr>
<tr>
<td>UTIL_HEAP_SZ</td>
<td>Utilities heap size (4 KB)</td>
<td>5000</td>
</tr>
<tr>
<td>LOCKLIST</td>
<td>Maximum storage for lock list (4 KB)</td>
<td>5000&lt;sup&gt;a&lt;/sup&gt;</td>
</tr>
<tr>
<td>APP_CTL_HEAP_SZ</td>
<td>Maximum applications control heap size (4 KB)</td>
<td>152</td>
</tr>
<tr>
<td>SORTHEAP</td>
<td>Sort list heap (4 KB)</td>
<td>32&lt;sup&gt;b&lt;/sup&gt;</td>
</tr>
<tr>
<td>STMTHEAP</td>
<td>SQL statement heap (4 KB)</td>
<td>8192</td>
</tr>
<tr>
<td>APPLHEAPSZ</td>
<td>Default application heap (4 KB)</td>
<td>2500</td>
</tr>
<tr>
<td>PCKCACHESZ</td>
<td>Package cache size (4 KB)</td>
<td>2048</td>
</tr>
<tr>
<td>STAT_HEAP_SZ</td>
<td>Statistics heap size (4 KB)</td>
<td>8000</td>
</tr>
<tr>
<td>MAXLOCKS</td>
<td>Percentage of lock lists per application</td>
<td>20</td>
</tr>
<tr>
<td>LOCKTIMEOUT</td>
<td>Lock timeout (sec.)</td>
<td>300 or higher</td>
</tr>
<tr>
<td>CHNGPGS_THRESH</td>
<td>Changed pages threshold</td>
<td>60</td>
</tr>
<tr>
<td>NUM_IOCLEANERS</td>
<td>Number of asynchronous page cleaners</td>
<td>Number of CPUs</td>
</tr>
<tr>
<td>NUM_IOSERVERS</td>
<td>Number of I/O servers</td>
<td>Number of disks</td>
</tr>
<tr>
<td>INDEXSORT</td>
<td>Index sort flag</td>
<td>YES</td>
</tr>
<tr>
<td>SEQDETECT</td>
<td>Sequential detect flag</td>
<td>YES</td>
</tr>
<tr>
<td>DFT_PREFETCH_SZ</td>
<td>Default prefetch size (4 KB)</td>
<td>128</td>
</tr>
<tr>
<td>LOGRETAIN</td>
<td>Sequential or circular log files</td>
<td>YES&lt;sup&gt;c&lt;/sup&gt;</td>
</tr>
</tbody>
</table>

<sup>a</sup> Maximum limit 1000000<br>
<sup>b</sup> Maximum limit 65536<br>
<sup>c</sup> Default is YES
Creating required NT user accounts

Create specific administrative and user accounts as follows:

1. Select **Start->Programs->Administrative Tools(Common)->User Manager for Domains** and copy the Administrator account as SIEBEL. This should create the SIEBEL account with all the needed privileges.

   Uncheck the USER MUST CHANGE PASSWORD box.

   Note that SIEBEL must be capitalized in order for the grantusr.sql script to run correctly later when adding the Siebel Database Server.

2. Select **User->New Local Group, sse_role**, and add the SADMIN and SIEBEL accounts to it.

Creating the Siebel File System

In addition to the RDBMS, the Siebel application requires a file system to hold objects such as user documents and spread sheets. To create the directory, open a DOS Command Window and enter:

<table>
<thead>
<tr>
<th>MAXAPPLS</th>
<th>Maximum number of active applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVG_APPLS</td>
<td>Maximum DB files open per application</td>
</tr>
<tr>
<td>MAXFILOP</td>
<td>Maximum DB files sopped per application</td>
</tr>
<tr>
<td>LOGFILSIZ</td>
<td>Log file size (4 KB)</td>
</tr>
<tr>
<td>LOGPRIMARY</td>
<td>Number of primary log files</td>
</tr>
<tr>
<td>LOGSECOND</td>
<td>Number of secondary log files</td>
</tr>
<tr>
<td>SOFTMAX</td>
<td>Percent log file reclaimed before soft checkpoint</td>
</tr>
</tbody>
</table>

- Based on the number of users plus at least 20 for Application Server connections
- Depends on the environment
- 500
- 8000
- 100
- 10
- 80

a. The setting should never be smaller than this, but may be increased.
b. Recommended size; this may increase or decrease depending on the amount of memory in the database server machine and the size of the database.
c. Setting this will cause the database to go into “Backup Pending” state. We recommend leaving this parameter set to the default.
To make the file system sharable in read/write, do the following:

1. Navigate to Start->Programs->Windows NT Explorer, right-click the new siebfile folder and select Sharing....
2. Click Shared As:.
3. Click Permissions and verify that Type of Access: is Full Control.
4. If it is not, change it to Full Control.
5. Click OK, Apply, and then OK again.

Installing the Siebel Gateway Server

Next, we installed the Siebel Gateway Server and started the Siebel Gateway Name Server Service.

Installing the Siebel Gateway Server

To install the Siebel Gateway Server, do the following:

1. Insert the Siebel Windows Server Programs CD-ROM.
2. Select Start->Programs->Windows NT Explorer, navigate to the CD-ROM directory \gtwysrvr, and double-click Setup.exe to start the installation program.
3. Read the Welcome to Gateway Server Setup screen and click Next > to continue.
4. A notice appears that Resonate Central Dispatch is not installed. Click Yes to continue installing without Resonate support.
5. Change your directory, if desired, and select Next >. For this example we installed Siebel 2000 on the d: drive, so we specified d:\sea621\gtwysrvr as the target directory.
6. Enter the Gateway Server NT Account Information, which is the NT account that will be used to run the NT service. We provided the following values:

<table>
<thead>
<tr>
<th>Account</th>
<th>SADMIN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Password</td>
<td>SADMIN</td>
</tr>
<tr>
<td>Password (retype)</td>
<td>SADMIN</td>
</tr>
</tbody>
</table>

7. Click Next >.
8. When the Gateway Server NT Services screen appears, click **Next** > to accept the default to start the service automatically.

9. When the Start Copying Files screen appears, click **Next** > to start copying the files.

10. When the Event Log screen appears, review the contents of the Event Log. Look for the message *GtwyNS()NT Service successfully installed*.

11. Click **Next** >.

12. The Restarting Windows screen appears.

13. Select **No, I will restart my computer later** and click **OK**.

At this point, we installed the Siebel 6.2.1.7 patch for Siebel Gateway Server from the distribution CD and followed the installation instructions (execute setup).

**Starting the Siebel Gateway Server NT service**

We choose not to reboot. To start the Gateway service, do the following:

1. Select **Start->Settings->Control Panel**.

2. Double-click **Services**.

3. Select the **Siebel Gateway Name Server** Service.

4. Click **Start**.

5. The status should be changed to Started.

6. Click **Close** to exit.

**Inspecting the Gateway Server directory**

To inspect the gateway directory, do the following:

1. Verify that the `d:\sie621\gtwysrvr` directory is created.

2. Inspect the contents of the GTWsetup.log file located in the `d:\sea621\gtwysrvr` directory.

3. Verify that there are many `.dll` files in the `d:\sea621\gtwysrvr\bin` directory.

   If you use NT Explorer to do this verification, be sure you have “display all files” enabled.
Installing the Siebel Server

In this section, we describe how to create the Siebel Enterprise Server, start the Siebel Enterprise Server NT Service, inspect the server directory, and verify the Siebel Server ODBC Data Source.

Installing the Siebel Server

To install the Siebel Server, make sure that the Siebel Gateway Server is installed and running by doing the following:

1. Select Start->Programs->Windows NT Explorer, navigate to the CD-ROM directory \siebsrvr, and double-click Setup.exe to launch the installation program.
2. Read the Welcome to the Siebel Server Installer screen and click Next >.
3. A notice appears indicating that Resonate Central Dispatch is not installed. Click Yes to continue installing without Resonate support.
4. When the Start the Siebel Gateway Server screen appears, click Next > since the Siebel Gateway Server is already started.
5. The Setup Type screen appears; select Typical to perform a typical installation and click Next >.
6. We specify d:\sea621\siebsrvr as the target directory.
7. The Gateway Server Address screen appears; click Browse and select ITSONT0 (our machine name) to specify the Gateway Server Address, then click OK.
8. Click Next >.
9. The Enterprise Server Information screen appears; accept the default parameters as shown:

<table>
<thead>
<tr>
<th>Name</th>
<th>siebel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Siebel Enterprise Server</td>
</tr>
</tbody>
</table>
10. Click Next >.
11. When asked if you want to create the Enterprise Server (siebel), click Yes.
12. The Siebel File System Location screen appears; click Browse, type in d:\siebfile, and click OK to specify the Siebel File System Location.
13. Click Next >.
14. The Server Database screen appears. Make sure that IBM DB2 Universal Database 6.1 is selected and then click Next >.
15. The Database Identification screen appears, we specified the database identification as shown:

<table>
<thead>
<tr>
<th>Database alias</th>
<th>siebeldb</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table Owner</td>
<td>SIEBEL</td>
</tr>
</tbody>
</table>

16. Click **Next >**.

17. The Server Database Account Login Information screen appears; specify the database login information as shown:

```
<table>
<thead>
<tr>
<th>User Name:</th>
<th>SADMIN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Password:</td>
<td>SADMIN</td>
</tr>
<tr>
<td>Confirm Password</td>
<td>SADMIN</td>
</tr>
</tbody>
</table>
```

18. Click **Next >**.

19. The Siebel Server screen appears; specify the server information as shown:

```
<table>
<thead>
<tr>
<th>Siebel Server Name</th>
<th>&lt;Machine Name&gt; (for example: ITSONT0)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Siebel server Description</td>
<td>Siebel Server</td>
</tr>
</tbody>
</table>
```

20. Click **Next >**.

21. The Siebel Server NT Service screen appears; click **Next >** to accept the default to start the Siebel Server NT Service Automatically.

22. The Synchronization Port Assignment screen appears; click **Next >** to accept the default of 40400 for the Synch Manager Port.

23. The eBriefings and eContent Services screen appears; select **Neither** and click **Next >** to turn off eBriefings and eContent Services.

24. The Java Thin Client Help File URL screen appears; type in 
   http://www.siebel.com/JavaClient/help/start.html and click **Next >**.

25. The Start Copying Files screen appears; click **Next >**.

26. The Enterprise Server “siebel” has been successfully created... screen appears; click **OK**.

27. When the Event Log screen appears, verify that there are no errors and click **Next >**.

28. When the Setup Complete screen appears, select **No I will restart my computer later** and click **OK**.

**Inspecting the Siebel Server directory**

Perform a verification as follows:
1. Verify that the d:sie621\siebsrvr directory is created.
2. Inspect the contents of the SVRsetup.log file located in the d:\sea621\siebsrvr directory.
3. Verify that there are many .dll files in the d:\sea621\siebsrvr\bin directory.

At this point we installed the Siebel 6.2.1.7 server patch (siebsrvr) from the distribution CD and followed the installation instructions (execute setup).

**Starting the Siebel Server NT service**

Start the service as follows:
1. Select Start->Programs->Control Panel.
2. Double-click Services.
3. Select the Siebel Server [siebel_ ITSONT0] service.
4. Click Start.
5. The status should be changed to Started.
6. Click Close to exit.

**Verifying the Siebel Server ODBC data source**

The Siebel Server installation program automatically creates the ODBC system data source name (DSN) that it uses to connect to the Siebel Database Server; verify this by doing the following:
1. Select Start->Settings->Control Panel.
2. Double-click ODBC Data Sources.
3. Click the System DSN tab.
4. Select SiebSrvr_siebel and click Configure....
5. The DB2 Message screen appears asking if you want to connect to the database; click Yes.
6. Enter SADMIN for the User ID and Password when prompted and click OK.
7. The DB2 Message screen appears indicating that the connection completed successfully; click OK.
8. Make sure that CLI/ODBC Settings - siebeldb are set as shown:

<table>
<thead>
<tr>
<th>Data source name</th>
<th>SiebSrvr_siebel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Siebel</td>
</tr>
</tbody>
</table>
Appendix A. Implementing the Siebel source environment on NT

Installing the Siebel Database Server

In this section, we describe how to install the Siebel Database Server code, execute grantusr.sql to create a Siebel Administrator (sadmin) and Database Owner (SIEBEL) of the Siebel database, configure the Siebel DB2 Database instance, and install the stored procedures code.

Installing the Siebel Database Server scripts

To install these scripts, do the following:

1. Select Start->Programs->Windows NT Explorer and navigate to the CD-ROM directory \dbsrvr; double-click Setup.exe to begin the installation.

2. Read the Welcome to Database Scripts Setup screen and click Next > to continue with the setup program.

3. Make sure that Typical is selected, change the destination directory to d:\sea621\dbsrvr, if necessary, and click Next >.

4. The Event Log screen appears; click Next >.

5. Setup completed successfully should appear on the Setup Complete screen.

6. Click Finish.

7. Install the Siebel 6.2.1.7 database server patch from the distribution CD and follow the installation instructions (execute setup). To commit the changes, we executed forceit.bat in a DB2 Command Window.

Granting database privileges

Next, to grant privileges, do the following:

1. Navigate to Start->Programs->DB2 for Windows NT->Command Window.

2. Issue the following commands:

   DB2 CONNECT TO siebeldb USER sadmin USING sadmin

   User ID | SIEBEL
   Password | SIEBEL

9. Click OK.

10. Click OK.
Configuring the Siebel DB2 instance

To configure the DB2 instance, do the following:

2. Highlight the DB2-DB2 service and press Startup....
3. In the Log On As: box, press This Account:
4. Choose the local computer (ITSONT0) in the List Names From: pick list.
5. Highlight SIEBEL, click Add, and click OK set the Password and Confirm password fields to SIEBEL, and then click OK.
6. Click Close.

Installing the stored procedures code

Now transfer and install the user-defined functions (UDFs) and stored procedures on the Siebel Database Server as follows:

2. In the DB2 Command Window, navigate to the d:\sea621\dbsrvr\db2udb\siebproc\winnt directory, type installsiebel, and click Enter. Follow the screen prompts to install the files.

Initializing the Siebel database

Next, we created the Siebel schema (tables and columns) in the precreated skeleton Siebel database and imported seed data; edited and ran the imprep.ksh script to import object definitions data (Siebel repository) into the procreated Siebel database; and populated the procreated Siebel File System with the required correspondence templates and Siebel Marketing files.

Creating the Siebel schema

Creating the Siebel schema implies the following:

- Edit, then run the install and imprepb batch files to set all variables and create the tables and indexes needed to set up the database server. Modify the specified parameters below.
- Use Notepad or an appropriate editor to edit the install.ksh script in the d:\sea621\dbsrvr\db2udb directory with the appropriate names for our installation, as follows:
SRC_USR = SADMIN
SRC_PSWD = SADMIN
SRC_TBLO = SIEBEL
SRC_TBLO_PSWD = SIEBEL
ODBC = SiebSrvr_siebel
DBSRVR_ROOT = d:/sea621/dbsrvr
DATA_AREA = “SIEBEL_4K”
INDX_AREA = “SIEBEL_IDX”
LONG_AREA = “SIEBEL_4KL”
DB16K_AREA = “SIEBEL_16K”
DATABASE_PLATFORM = DB2udb
DB_LANG = enu
DB_UNICODE_FLG = N
DBSERVER_OS = winnt
SIEBEL_HOME = d:/sea621/siebsrvr

To accomplish this, do the following:

1. Make sure that the line SRC_USR user SADMIN is in uppercase. This is recommended by Siebel Systems.
2. Make sure that the line SRC_PSWD user password SADMIN is in uppercase. This is recommended by Siebel Systems.
3. Make sure that the line SRC_TBLO table owner user SIEBEL is capitalized.
4. Make sure that the line SRC_TBLO_PSWD table owner password SIEBEL is capitalized.
5. Specify ODBC = SiebSrvr_siebel.
7. Specify DATA_AREA = “SIEBEL_4K”.
8. Specify INDX_AREA = “SIEBEL_IDX”.
10. Specify DB16K_AREA = “SIEBEL_16K”.
11. Place the text pound sign “#” (without quotes) in front of the line DATABASE_PLATFORM = Unspecified.
12. Remove the text “#” from the line # DATABASE_PLATFORM = DB2udb.
13. Place the text “#” (without quotes) in front of the line DB_LANG = Unspecified.
14. Remove the text “#” from the line DB_LANG = enu.
15. Place the text “#” (without quotes) in front of the line DB_UNICODE_FLG = Unspecified.
16. Remove the text “#” from the line DB_UNICODE_FLG = N.
17. Place the text “#” (without quotes) in front of the line `DBSERVER_OS = Unspecified`.

18. Remove the text “#” from the line `DBSERVER_OS = winnt`.

19. Specify `SIEBEL_HOME = d:/sea621/siebsrvr`.

20. Save the edited `install.ksh` file.

21. Select **Start->Programs->Windows NT Explorer**, navigate to the database platform subdirectory `d:\sea621\dbsrvr\db2udb`, and double-click **sh.exe** (this opens a Korn shell window).

22. In the Korn shell window, type `. /install.ksh` and press **Enter**.

23. Press **Y** and **Enter** if the values are correct (type any other key to terminate if the values are incorrect).

24. The `install.ksh` command should run approximately 45 minutes. It **must** run in this directory, or it will not set all the environment variables.

   **Note:** You must **finish** running `install.ksh` before you run the `imprep.ksh` command file.

   Figure A-2 shows some error messages we received. W_ tables are not yet created at this point of the installation. We therefore got error messages indicating they do not exist. It is safe to ignore those error messages.

   ![Error messages](image)

25. Type `exit` and press **Enter** to close the Korn shell window.
Verifying the results of install.ksh

To verify the results of the installation, do the following:

1. Select Start->Programs->DB2 for Windows NT->Command Center.
2. In the interactive window, enter:
   ```
   connect to siebeldb user sadmin using sadmin
   ```
3. Click Script->Execute or the gears icon at the top; the result should be that you are connected to the database.
4. In the interactive window, enter:
   ```
   select count(*) from SIEBEL.S_ZIPCODE
   ```
5. Click Script->Execute or the gears icon at the top; the result displayed should be approximately 42785.
6. Select Results->Shutdown DB2 tools to close the Command Center.

Importing the Siebel object definitions

To import the object definitions, we used Notepad to edit the imprep.ksh script in the d:\sea621\dbsrvr\db2udb directory as follows:

```
SRC_USR = SADMIN
SRC_PSWD = SADMIN
SRC_TBLO = SIEBEL
ODBC =SiebSrvr_siebel
REPOS_NAME = "Siebel Repository"
DBSRVR_ROOT = d:\sea621\dbsrvr
DATABASE_PLATFORM = DB2udb
DB_LANG = enu
SIEBEL_HOME = d:\sea621\siebsrvr
```

To accomplish the above, do the following:

1. Make sure that the line SRC_USR user SADMIN is in uppercase.
2. Make sure that the line SRC_PSWD user password SADMIN is in uppercase.
3. Make sure that the line SRC_TBLO table owner user SIEBEL is capitalized.
5. Specify REPOS_NAME = "Siebel Repository".
6. Specify DBSRVR_ROOT = d:\sea621\dbsrvr.
7. Place the text pound sign “#” (without quotes) in front of the line DATABASE_PLATFORM = Unspecified.
8. Remove the text “#” from the line DATABASE_PLATFORM = DB2udb.
9. Place the text “#” (without quotes) in front of the line DB_LANG = Unspecified.
10. Remove the text “#” from the line `DB_LANG = enu`.

11. Specify `SIEBEL_HOME = d:\sea621\siebsrvr`.

12. Save the edited `imprep.ksh` file.

13. Navigate to the database platform subdirectory and double-click `sh.exe` (this opens a Korn shell window).

14. In the Korn shell window, type `./imprep.ksh` and press `Enter`.

15. Press `Y` and `Enter` if the values are correct (type any other key to terminate if the values are incorrect).

16. The `imprep.ksh` command should run approximately 20 minutes.

17. Type `exit` and press `Enter` to close the Korn shell window.

**Verifying the results of `imprep.ksh`**

To verify that `imprep.ksh` ran successfully, do the following:

1. Select `Start->Programs->DB2 for Windows NT->Command Center`.

2. In the interactive window, enter:
   ```
   connect to siebeldb user sadmin using sadmin
   ```

3. Click `Script->Execute` or the gears icon at the top; the results should be that you are connected to the database.

4. In the interactive window, enter:
   ```
   select count(*) from SIEBEL.S_VIEW
   ```

5. Click `Script->Execute` or the gears icon at the top; the results displayed should be approximately 1780.

6. Select `Results->Shutdown DB2 tools` to close the Command Center.

**Populating the Siebel File System**

Specific files needed to run the Siebel File System are provided with the Siebel Database Server software. These files need to be manually moved to the subdirectory named files, which is created automatically when you install the Siebel Database Server. You must populate the File System directory with these files after installing the Database Server, and before running the Siebel Client.

To populate the file system directory, do the following:

1. Select `Start->Programs->Windows Explorer`, navigate to the `d:\sea621\dbsrvr\files` directory, and copy all the files to the File System `d:siebfile` directory.

2. Verify that the files are where they need to be.
Verifying the install

Next, install the Siebel Client and log on to the Siebel Call Center Demo to verify that the installation is successful.

Installing the Siebel client

Install the Siebel client as follows:

1. Select Start->Programs->Windows NT Explorer, navigate to the \sea subdirectory on the Siebel Client Installation CD-ROM, and double-click Setup.exe to begin the installation.

2. When the Welcome to Client Setup screen appears, click Next > to begin the installation.

3. Make sure that Typical is selected, change the destination directory to d:\sea\client, and click OK.

4. Click Next >.

5. When the Server Database screen appears, make sure that IBM DB2 Universal Database 6_1 is selected and click Next >.

6. Click NEXT (take default) when the Document Integrator question displays.

7. Click NEXT (take default) when the ODBC Pack question displays.

8. Click NEXT.

9. On the Server Locations screen, Siebel Remote Server should be set to blank and Siebel File System should be set to d:\siebfile.

10. Click Next >.

11. On the Database Identification screen, the Database Alias should be set to siebeldb and the Table Owner should be set to SIEBEL.

12. Click Next >.

13. On the Enterprise Server Information screen, the Gateway Server Address should be set to the machine name of the Siebel Gateway Server (ITSONT0). The Enterprise Server should be set to siebel (this corresponds to the name used when you installed Siebel Enterprise Server.

14. Click Next >.

15. On the Server Request Information screen, accept the defaults of SRMSynch for the Request Component and blank out (clear out the field) the Request Server Name.

16. Click Next >.
17. On the Analysis Proxy Server Setup, the Siebel Server Name should be set to blank (clear out field).

18. Click Next >.

19. On the Fulcrum Setup window, the Search Hostname and Search Port Number fields should be blank (clear out the fields). The Search Definition Name can remain as the default of Fulcrum Sample.

20. Click Next >.

21. On the Select Program Folder screen, click Next > to accept the default program folder.

Note: You may see several Self-registration error messages during the installation; ignore them.

After the install is completed, an Event Log screen is displayed. Scroll through the log to view the events.

In the Event Log screen you see the following messages, which are expected:
   - The Fulcrum ODBC Driver Software component was not found
   - The IdCentric Software component was not found

To view the event log using Notepad, you can also go to d:\sea\client and view the SEAsetup.err and SEAsetup.log files.

Click Next > after viewing the Event Log window.

The Registry Log screen is displayed. Scroll through to view and do as follows:
1. Click Next > when done.
2. Click Finish (Restart system later).
3. Install the Siebel Client patch version 6.2.1.7.
4. Select Start->Programs->Control Panel->Services and stop and start the Siebel Enterprise Server.

Testing the installation using the Siebel Call Center

To test the installation, use the Call Center as follows:

1. Select Start->Programs->Siebel Client 6.0->Siebel Call Center.
2. Agree with the licensing terms.
3. Log in with a User name of SADMIN and the Password SADMIN. For the Connect to drop-down, select the Server option.
4. Click OK.
5. The first time you log in, an error window displays the message: No valid license keys were found in the database. Please enter a license key in the following dialog, or contact your systems administrator.

6. Click OK.

7. On the License Keys window press Add Key and enter the license key found on the back of the CD case.

8. Click OK.

9. Press OK again on the License Key window that displays the current license keys with a status of OK on the key just entered.


**Installing Siebel Tools**

Next, we install the Siebel Tools.

**Detailed instructions**

To install the Siebel Tools, we do the following:

1. Select Start->Programs->Windows NT Explorer, navigate to the \seaTOOLS subdirectory on the Siebel Client Installation CD-ROM, and double-click Setup.exe.

2. When the Welcome to Client Setup screen appears, press Next > to start the installation.


4. On the Server Database screen, make sure that IBM DB2 Universal database 6_1 is selected and press Next >.

5. We choose not to install the ODBC pack.

6. On the Servers Location screen, set the Siebel Remote Repository to blank, the Siebel File System to d:\siebfile, and press Next >.

7. On the Database Identification screen, set the Database Alias to siebeldb, the Table Owner to SIEBEL, and click Next >.

8. Click Next > to accept the default Program Folder (i.e., Siebel Tools 6.0).

10. On the **Setup Complete** screen, a message displays the setup has completed successfully. Click **Finish** to exit.

11. Install Siebel Tools patch version 6.2.1.7 from the distribution CD and follow the installation instructions (execute setup).

**Testing Siebel Tools**

To test the Siebel Tools, we do the following:

1. Select **Start->Programs->Siebel Tools 6.0->Siebel Tools**.
2. Agree with the licensing terms.
3. Log in with a User name of SADMIN and the Password SADMIN. For the Connect to drop-down, select the **Server** option.
4. Click **OK**.
5. The first time we log in, an error window displays the message indicating No valid license keys were found in the database. Please enter a license key in the following dialog, or contact your systems administrator.
6. Click **OK**.
7. On the License Keys window press **Add Key** and enter the license key found on the back of the CD case.
8. Click **OK**.
9. Press **OK** again on the License Key window that displays the current license keys with a Status of OK on the key just entered.
10. The Siebel Tools explorer window displays.

**Creating sample data**

The supplied sample Siebel database is an SQL Anywhere database. There is no supplied tool to move this sample data into the database you create in the installation phase. You have to migrate it, which we did later on in the Oracle for NT environment. For this first test in the DB2 for NT environment, we keyed directly into the DB2 database some sample data using Siebel Call Center. What we did is documented in “Entering data using Siebel Call Center” on page 121.
The .bat files

**db2_db_mgr_cfg.bat**

```
DB2 UPDATE DATABASE CONFIGURATION FOR siebeldb USING DFT_DEGREE 1
DB2 UPDATE DATABASE CONFIGURATION FOR siebeldb USING DFT_QUERYOPT 5
DB2 UPDATE DATABASE CONFIGURATION FOR siebeldb USING DBHEAP 7429
  pause
DB2 UPDATE DATABASE CONFIGURATION FOR siebeldb USING CATALOGCACHE_SZ 5558
DB2 UPDATE DATABASE CONFIGURATION FOR siebeldb USING LOGBUFSZ 512
DB2 UPDATE DATABASE CONFIGURATION FOR siebeldb USING UTIL_HEAP_SZ 5000
  pause
DB2 UPDATE DATABASE CONFIGURATION FOR siebeldb USING LOCKLIST 5000
DB2 UPDATE DATABASE CONFIGURATION FOR siebeldb USING APP_CTL_HEAP_SZ 152
DB2 UPDATE DATABASE CONFIGURATION FOR siebeldb USING SORTHEAP 20000
  pause
DB2 UPDATE DATABASE CONFIGURATION FOR siebeldb USING STMTHEAP 8192
DB2 UPDATE DATABASE CONFIGURATION FOR siebeldb USING APPLHEAPSZ 2500
DB2 UPDATE DATABASE CONFIGURATION FOR siebeldb USING PCKCACHESZ 2048
  pause
DB2 UPDATE DATABASE CONFIGURATION FOR siebeldb USING STAT_HEAP_SZ 8000
DB2 UPDATE DATABASE CONFIGURATION FOR siebeldb USING MAXLOCKS 20
DB2 UPDATE DATABASE CONFIGURATION FOR siebeldb USING LOCKTIMEOUT 300
  pause
DB2 UPDATE DATABASE CONFIGURATION FOR siebeldb USING CHNGPGS_THRESH 60
DB2 UPDATE DATABASE CONFIGURATION FOR siebeldb USING NUM_IOCLEANERS 1
DB2 UPDATE DATABASE CONFIGURATION FOR siebeldb USING NUM_IOSERVERS 1
  pause
```
DB2 UPDATE DATABASE CONFIGURATION FOR siebeldb USING INDEXSORT YES
DB2 UPDATE DATABASE CONFIGURATION FOR siebeldb USING SEQDETECT YES
DB2 UPDATE DATABASE CONFIGURATION FOR siebeldb USING DPT_PREFETCH_SZ 128
DB2 UPDATE DATABASE CONFIGURATION FOR siebeldb USING LOGRETAIN NO
DB2 UPDATE DATABASE CONFIGURATION FOR siebeldb USING MAXAPPLS 25
DB2 UPDATE DATABASE CONFIGURATION FOR siebeldb USING AVG_APPLS 15
DB2 UPDATE DATABASE CONFIGURATION FOR siebeldb USING MAXFILOP 500
DB2 UPDATE DATABASE CONFIGURATION FOR siebeldb USING LOGFILSIZ 8000
DB2 UPDATE DATABASE CONFIGURATION FOR siebeldb USING LOGPRIMARY 10
DB2 UPDATE DATABASE CONFIGURATION FOR siebeldb USING LOGSECOND 100
DB2 UPDATE DATABASE CONFIGURATION FOR siebeldb USING SOFTMAX 80
DB2 FORCED APPLICATIONS ALL
DB2 DB2STOP
DB2 DB2START

db2_db2set.bat

DB2SET DB2_HASH_JOIN=NO
DB2SET DB2_RR_TO_RS=YES
DB2SET DB2_MMAP_WRITE=OFF
DB2SET DB2_MMAP_READ=OFF
DB2SET DB2_CORRELATED_PREDICATES=YES

db2_db_cfg.bat

DB2 UPDATE DATABASE CONFIGURATION FOR siebeldb USING DFT_DEGREE 1
DB2 UPDATE DATABASE CONFIGURATION FOR siebeldb USING DFT_QUERYOPT 5
DB2 UPDATE DATABASE CONFIGURATION FOR siebeldb USING DBHEAP 7429
DB2 UPDATE DATABASE CONFIGURATION FOR siebeldb USING CATALOGCACHE_SZ 5558
DB2 UPDATE DATABASE CONFIGURATION FOR siebeldb USING LOGBUFSZ 512
DB2 UPDATE DATABASE CONFIGURATION FOR siebeldb USING UTIL_HEAP_SZ 5000
DB2 UPDATE DATABASE CONFIGURATION FOR siebeldb USING LOCKLIST 5000
DB2 UPDATE DATABASE CONFIGURATION FOR siebeldb USING APP_ctl_HEAP_SZ 152
DB2 UPDATE DATABASE CONFIGURATION FOR siebeldb USING SORTHEAP 20000
DB2 UPDATE DATABASE CONFIGURATION FOR siebeldb USING STMTHEAP 8192
DB2 UPDATE DATABASE CONFIGURATION FOR siebeldb USING PCKCACHESZ 2048

118 Migrating Siebel Database from DB2/Oracle for NT to DB2 for OS/390
Appendix B. The .bat files

DB2 UPDATE DATABASE CONFIGURATION FOR siebeldb USING STAT_HEAP_SZ 8000
DB2 UPDATE DATABASE CONFIGURATION FOR siebeldb USING MAXLOCKS 20
DB2 UPDATE DATABASE CONFIGURATION FOR siebeldb USING LOCKTIMEOUT 300
pause
DB2 UPDATE DATABASE CONFIGURATION FOR siebeldb USING CHNGPGS_THRESH 60
DB2 UPDATE DATABASE CONFIGURATION FOR siebeldb USING NUM_IOCLEANERS 1
DB2 UPDATE DATABASE CONFIGURATION FOR siebeldb USING NUM_IOSERVERS 1
pause
DB2 UPDATE DATABASE CONFIGURATION FOR siebeldb USING INDEXSORT YES
DB2 UPDATE DATABASE CONFIGURATION FOR siebeldb USING SEQDETECT YES
DB2 UPDATE DATABASE CONFIGURATION FOR siebeldb USING DFT_PREFETCH_SZ 128
pause
DB2 UPDATE DATABASE CONFIGURATION FOR siebeldb USING LOGRETAIN NO
DB2 UPDATE DATABASE CONFIGURATION FOR siebeldb USING MAXAPPLS 25
DB2 UPDATE DATABASE CONFIGURATION FOR siebeldb USING AVG_APPLS 15
pause
DB2 UPDATE DATABASE CONFIGURATION FOR siebeldb USING MAXFILOP 500
DB2 UPDATE DATABASE CONFIGURATION FOR siebeldb USING LOGFILSZ 8000
DB2 UPDATE DATABASE CONFIGURATION FOR siebeldb USING LOGPRIMARY 10
pause
DB2 UPDATE DATABASE CONFIGURATION FOR siebeldb USING LOGSECOND 100
DB2 UPDATE DATABASE CONFIGURATION FOR siebeldb USING SOFTMAX 80
pause
DB2 FORCE APPLICATIONS ALL
DB2 DB2STOP
DB2 DB2START

### tablespace.bat

rem **********************************************************
rem * If you installed your siebel DB2 database using other than "db2admin" you
rem * will have to edit the next line of this bat file accordingly.
rem **********************************************************
rem DB2 CONNECT TO siebeldb USER db2admin USING db2admin
db2 connect to siebeldb user sadmin using sadmin2
rem **********************************************************
rem * If you installed on a drive other than "D" or into a directory other than
rem * "SQLLIB" you will have to edit the remaining CREATE lines of this bat
rem * file accordingly.
rem **********************************************************
DB2 CREATE REGULAR TABLESPACE SIEBEL_4K PAGESIZE 4 K MANAGED BY DATABASE USING
(FILE 'D:\SQLLIB\SIEBEL_4K' 64000) EXTENTSIZE 16 OVERHEAD 24.1 PREFETCHSIZE 8
TRANSFERRATE 0.9 BUFFERPOOL IBMDEFAULTBP
pause
DB2 CREATE REGULAR TABLESPACE SIEBEL_16K PAGESIZE 16 K MANAGED BY DATABASE USING
(FILE 'D:\SQLLIB\SIEBEL_16K' 64000) EXTENTSIZE 16 OVERHEAD 24.1 PREFETCHSIZE 8 TRANSFERRATE 0.9 BUFFERPOOL BUF16K
pause
DB2 CREATE LONG TABLESPACE SIEBEL_4KL PAGESIZE 4 K MANAGED BY DATABASE USING
(FILE 'D:\SQLLIB\SIEBEL_4KL' 25600) EXTENTS SIZE 16 OVERHEAD 24.1 PREFETCH SIZE 8
TRANSFERRATE 0.9 BUFFERPOOL IBMDEFAULTBP
pause
DB2 CREATE REGULAR TABLESPACE SIEBEL_IDX PAGESIZE 4 K MANAGED BY DATABASE USING
(FILE 'D:\SQLLIB\SIEBEL_IDX' 80000) EXTENTS SIZE 16 OVERHEAD 24.1 PREFETCH SIZE 8
TRANSFERRATE 0.9 BUFFERPOOL IBMDEFAULTBP
pause
DB2 CREATE TEMPORARY TABLESPACE TEMP4K PAGESIZE 4 K MANAGED BY SYSTEM USING
('D:\SQLLIB\TEMP4K') EXTENTS SIZE 32 OVERHEAD 24.1 PREFETCH SIZE 128 TRANSFERRATE
0.9 BUFFERPOOL BUF4KTEMP
pause
DB2 CREATE TEMPORARY TABLESPACE TEMP16K PAGESIZE 16 K MANAGED BY SYSTEM USING
('D:\SQLLIB\TEMP16K') EXTENTS SIZE 32 OVERHEAD 24.1 PREFETCH SIZE 128 TRANSFERRATE
0.9 BUFFERPOOL BUF16K
pause
DB2 CREATE TEMPORARY TABLESPACE TEMP32K PAGESIZE 32 K MANAGED BY SYSTEM USING
('D:\SQLLIB\TEMP32K') EXTENTS SIZE 32 OVERHEAD 24.1 PREFETCH SIZE 128 TRANSFERRATE
0.9 BUFFERPOOL BUF32KTEMP
pause
DB2 FORCE APPLICATIONS ALL

forceit.bat

DB2 FORCE APPLICATIONS ALL
DB2 DB2STOP
DB2 DB2START

bufferpool.bat

DB2 CONNECT TO siebeldb USER sadmin USING sadmin2
DB2 CREATE BUFFERPOOL BUF4KTEMP SIZE 25000 PAGESIZE 4 K
DB2 CREATE BUFFERPOOL BUF32KTEMP SIZE 1000 PAGESIZE 32 K
DB2 CREATE BUFFERPOOL BUF16K SIZE 1000 PAGESIZE 16 K
DB2 FORCE APPLICATIONS ALL.
Creating sample data

Entering data using Siebel Call Center

In order to have data to move for our project, we created a small number of records in the Siebel database on DB2/NT (for subsequent migration). See the following figures for examples of the user data we entered. You may wish to enter a similar amount of data, which was: five account records, 12 contact records, seven activity records, five invoice records and one opportunity record.

Figure C-1: Entering Account and Contact user data
### Figure C-2   Entering Activities user data

<table>
<thead>
<tr>
<th>New</th>
<th>Activity Type</th>
<th>Description</th>
<th>Due</th>
<th>Assigned To</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Appointment</td>
<td>meet with operation manager</td>
<td>3/16/01</td>
<td>SADMIN</td>
</tr>
<tr>
<td></td>
<td>Appointment</td>
<td>meet with purchasing manager</td>
<td>3/16/01</td>
<td>SADMIN</td>
</tr>
<tr>
<td></td>
<td>Call</td>
<td>New toys available</td>
<td>3/16/01</td>
<td>SADMIN</td>
</tr>
</tbody>
</table>

### Figure C-3   Entering Invoices user data

<table>
<thead>
<tr>
<th>Invoice Type</th>
<th>Invoice Code</th>
<th>Invoice Amount</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Receivable</td>
<td>Sales</td>
<td>$20,000.00</td>
<td>Open</td>
</tr>
<tr>
<td>Receivable</td>
<td>Service</td>
<td>$15,000.00</td>
<td>Open</td>
</tr>
</tbody>
</table>

### Figure C-4   Entering Opportunities user data

<table>
<thead>
<tr>
<th>Close Date</th>
<th>Opportunity</th>
<th>Account</th>
<th>Site</th>
</tr>
</thead>
<tbody>
<tr>
<td>3/16/01</td>
<td>new products available</td>
<td>ABC TOYS</td>
<td></td>
</tr>
</tbody>
</table>
For our EIM migration tests we created data in the Accounts table and Account Addresses table as shown in Figure C-5 and Figure C-6.

**Figure C-5  Entering data into Account table**

**Figure C-6  Manually entering Account Addresses**
Migrating the Siebel sample database

We first invoke ODBCSQL as shown in Figure C-7.

![Figure C-7 Invoking ODBCSQL](image)

Next, we spool the Siebel table names into the spool.txt file, as shown in Figure C-8.

![Figure C-8 Spooling the Siebel table names](image)
The spool.txt file is used as input for DATAEXP. Because this file is extracted from S_TABLE it is expected that DATAEXP will have errors (Figure C-9). S_TABLE represents the Siebel logical entities and not the physical schema.

**Figure C-9 Creating SQLANYSAMP.DAT**

Figure C-9 shows how we create the input SQLANYSAMP.DAT for DATAIMP.

Next, we need to generate a list of delete statements to delete the target data environment. We use ODBCSQL and, assuming the table owner is SIEBEL, we issue the following SQL statement:

```
select 'delete from SIEBEL.'||"TABLE_NAME"||';' from all_tables where owner = 'SIEBEL'
```

This statement creates a spool.txt file which needs to be edited and the headers deleted. Then, we need to connect to Oracle and run the delete statements.

We now have the Oracle database with all the tables empty.

The next step is to use DATAIMP to import the sample database into Oracle.

```
DATAIMP /U SIEBEL /P brlglte /C SiebSrvr_siebeloracle /D SIEBEL /F sqlanysamp.dat
```

The final step is to verify the migration using Siebel Call Center and check the migrated data is in the Oracle for NT database.
This appendix shows the DB2 for OS/390 customizations we did and describes the jobs, commands, and parameter settings we implemented. It includes the following sections:

- DSNZPARM parameters
- Creating STOGROUPs
- Creating the Siebel databases
- Creating the Siebel table spaces
- Creating Siebel tables and indexes
- Creating stored procedure objects
DSNZPARM parameters

```
//DB2DE JOB (999,POK), 'DB2 INSTALL', CLASS=A, MSGCLASS=T, NOTIFY=&SYSUID
/*JOBPARM L=9999,SYSAFF=SC04
************************************************************************************/
//* JOB NAME = DSN7IJUZ *
//* *
//* DESCRIPTIVE NAME = ZPARMS USED FOR SIEBEL MIGRATION PROJECT *
//* *
//* STATUS = VERSION 6 *
//* *
//* FUNCTION = DSNZPARM AND DSNHDECP UPDATES *
//* *
//* PSEUDOCODE = *
//* DSN7IZA STEP ASSEMBLE DSN6.... MACROS, CREATE DSNZPARM *
//* DSN7IZL STEP LINK EDIT DSNZPARM *
//* DSN7LOO STEP UPDATE PASSWORDS *
//* DSN7IZP STEP ASSEMBLE DSNHDECP DATA-ONLY LOAD MODULE *
//* DSN7IZQ STEP LINK EDIT DSNHDECP LOAD MODULE *
//* DSN7IMQ STEP SMP/E PROCESSING FOR DSNHDECP *
//* *
//* NOTES = STEP DSN7IMQ MUST BE CUSTOMIZED FOR SMP. SEE THE NOTES *
//* NOTES PRECEDING STEP DSN7IMQ BEFORE RUNNING THIS JOB. *
//* *
************************************************************************************/
//* *
//DSN7IZA EXEC PGM=ASMA90,PARM='OBJECT,NODECK'
//SYSLIB DD DISP=SHR,
  // DSN=DSN610.SDNSMACS
  // DSN=SYSL1.IMACLIB
//SYSLIN DD DSN=&LOADSET(DSN7ILMM), DISP=(NEW,PASS),
  // UNIT=SYSALLDA,
  // SPACE=(800,(50,50,2)), DCB=(BLKSIZE=800)
//SYSPRINT DD SYSOUT=* 
//SYSUDUMP DD SYSOUT=* 
//SYSUT1 DD UNIT=SYSALLDA, SPACE=(800,(50,50),,ROUND)
//SYSUT2 DD UNIT=SYSALLDA, SPACE=(800,(50,50),,ROUND)
//SYSUT3 DD UNIT=SYSALLDA, SPACE=(800,(50,50),,ROUND)
//SYSSIN DD *
  DSN6ENV MVS=XA
  DSN6SPRM RESTART, ALL,
  ABEXP=YES
  ABIND=YES,
  AUTH=YES,
```
AUTHCACH=1024
BINDNV=BINDADD,
BMPTOUT=4,
CACHEDYN=YES, <-IT WAS NO
CACHEPAC=32768,
CACHERAC=32768,
CATALOG=DB2V610D,
CDSRDEF=1,
CHGDC=NO,
CONTSTOR=NO,
DECDIV3=NO,
DEFLTID=IBMUSER,
DESCSTAT=NO,
DLITOUT=6,
DSMAX=21000, <- WAS 3000
EDMPOOL=14812,
EDMSPAC=0,
EDPROP=NO,
HOPAUTH=BOTH,
IRLMAUT=YES,
IRLMPRC=IRLDPROC,
IRLMPRC=IRLDPROC,
IRLSID=IRLD,
IRLMRWT=60,
IRLMSWT=300,
LEMAX=20,
MAXRBLK=4000,
MAXKEEPD=5000,
NUMLKT=1000,
NUMLKUS=1000, <- WAS 10000 (WAS 0 FOR SIEBEL1)
OPTHINTS=NO,
RECALL=YES,
RECALLD=120,
RELCURHL=YES,
RETLWAIT=0,
RETVLCFK=NO,
RGFCOLID=DSNRGCOL,
RGFBDBNAM=DSNRGFBDB,
RGFDDEPL=NO,
RGFDEFLT=ACCEPT,
RGFESC=*
RGFFULLQ=YES,
RGFINSTL=NO,
RGFNMOFT=DSN_REGISTER_OBJT,
RGFNMPRT=DSN_REGISTER_APPL,
RRULOCK=NO,
SEQCACH=BYPASS,
SEQPRES=NO,
SITETYP=LOCALSITE,
SRTPool=1000,
SYSADM=KARRAS,
SYSADM2=HAIMO,
SYSOPR1=SYSOPR,
SYSOPR2=SYSOPR,
TRKRSITE=NO,
UTIMOUT=6,
XLKUPOLT=NO

DSN6ARVP
ALCUNIT=BLK,
ARCWRTC=(1,3,4),
ARCWTR=ES,
ARCPFX1=DB2V610D.ArchLog1,
ARCPFX2=DB2V610D.ArchLog2,
ARCRETN=9999,
BLKSIZE=28672,
CATALOG=NO,
COMPACT=NO,
PRIQTY=1234,
PROTECT=NO,
QUIESCE=5,
SEQQTY=154,
TSTAMP=NO,
UNIT=TAPE,
UNIT2=

DSN6LOGP
DEALLC=(0),
MAXARCH=1000,
MAXRTU=2,
OUTBUFF=4000,
TWOACTV=YES,
TWOARCH=YES,
WRTHRSH=20,
ARC2FRST=NO

DSN6SYSP
AUDITST=NO,
BACKODUR=5,
CONDBAT=150000, <--- WAS 64
CTHREAD=70,
DBPROTCL=DRDA,
DLDREQ=5,
DSTIME=5,
EXTRAREQ=100,
EXTRASRV=100,
IDBACK=20,
IDFORE=40,
IDX8POOL=BP0,
LBACKOUT=AUTO,
LOBVALA=2048,
LOBVALS=2048,
LOGAPSTG=0,
LOGLOAD=50000,
Appendix D. DB2 customization jobs

MAXDBAT=500,  <--- WAS 64
MON=NO,
MSONSIZE=8192,
PCLOSEN=5,
PCLOSET=10,
RLF=NO,
RFLTB=120,
RFLERR=NOLIMIT,
RFAUTH=SYSIBM,
ROUTCDE=(1),
EXTSEC=YES,  <--- WAS NO
SMFACCT=(1),
SMFSTAT=YES,
STATIME=30,
STORMXAB=0,
STORPROC=DB2DSPAS,
STORTIME=180,
TBSBPOOL=BPO,
TRACSTR=NO,
TRACTBL=16,
URCHKTH=0,
WLMVW=

DSN6FAC   DDF=NO,
       CMTSTAT=INACTIVE,
       IDTHTOIN=0,
       RESYNC=2,
       RLFERR=NOLIMIT,
       TCPALVER=NO,
       MAXTYPE1=0,
       TCPKPAVL=ENABLE,
       POOLINAC=120

DSN6GRP   DSMSHARE=NO,
       GRPNAME=DSNCAT,
       MEMBNAME=DSN1,
       COORDNDT=NO,
       ASSIST=NO

END

//********************************************************************
//* LINK EDIT THE NEW DSNZPARM MEMBER. PUT LOAD MODULE IN SDSNEXIT.  *
//*********************************************************************/

//DSN6IL EXEC PGM=IEWL,PARM='LIST,XREF,LET,RENT',
   COND=(4,LT)
//ADSNLOAD DD DISP=SHR,
//DSN=DSN610.DSNLOAD
// DD DISP=SHR,
//DSN=DSN610.DSNLOAD
//SYSPUNCH DD DSN=&LOADSET(DSNTILMM),DISP=(OLD,DELETE)
//SYSLMD DD DISP=SHR,
//DSN=DB2V610D.SDSNEXIT
Migrating Siebel Database from DB2/Oracle for NT to DB2 for OS/390

#include SYSPUNCH(DSNTILMM)
#include ADSNLOAD(DSNZPARM)
ORDER DSNAA
#include ADSNLOAD(DSNAA)
#include ADSNLOAD(DSNFSYSP)
#include ADSNLOAD(DSNJARVP)
#include ADSNLOAD(DSNJLOGP)
#include ADSNLOAD(DSNTPARM)
#include ADSNLOAD(DSNWDIR1)
#include ADSNLOAD(DSNZMSTR)
#include ADSNLOAD(DSN3DIR1)
#include ADSNLOAD(DSN7GRP)
ENTRY DSNZMSTR
NAME DSNZDB2D(R)
/*
 */
/**
 **  CHANGE LOG INVENTORY:
 **  UPDATE BSDS
 **
/**
//DSNTLOG EXEC PGM=DSNJU003,COND=(4,LT)
//STEPLIB DD DISP=SHR,DSN=DSN610.SDSNLOAD
//SYSUT1 DD DISP=OLD,DSN=DB2Y610D.BSDS01
//SYSUT2 DD DISP=OLD,DSN=DB2Y610D.BSDS02
//SYSPRINT DD SYSOUT=* 
//SYSDUMP DD SYSOUT=*  
//SYSSIN DD *

#define DDF LOCATION=DB2D,LUNAME=SCPDB2D, 
      NOPASSWD,RESPORT=33323,PORT=33322 
/*
*******************************************************************************/
/**
 **  ASSEMBLE AND LINK EDIT DATA-ONLY LOAD MODULE DSNHDECP.
 **  THE FOLLOWING STEPS ARE NEEDED ONLY IF THE
 **  VALUES ARE CHANGED FROM THOSE WHICH ARE SHIPPED.
*******************************************************************************/
//DSNTIZP EXEC PGM=ASMA90,PARM='OBJECT,NODECK',COND=(4,LT)
//SYSLIB DD DISP=SHR, 
  // DSN=DSN610.SDSNMACS 
//SYSLIN DD DSN=A&LOADSET(DSNHDECA),DISP=(NEW,PASS),UNIT=SYSALLD 
  // SPACE=(80,(50,50,2)),DCB=(BLKSIZE=80) 
//SYSPRINT DD SYSOUT=* 
//SYSDUMP DD SYSOUT=*  
//SYSUT1 DD UNIT=SYSALLDA,SPACE=(800,(50,50),,,ROUND) 
//SYSUT2 DD UNIT=SYSALLDA,SPACE=(800,(50,50),,,ROUND) 
//SYSUT3 DD UNIT=SYSALLDA,SPACE=(800,(50,50),,,ROUND) 
//SYSSIN DD * 
DSNHDECM CHARSET=ALPHANUM,
ASCCSID=819,  <--- WAS 0
AMCCSID=65534,
AGCCSID=65534,
SCSCSID=37,
MCCSID=65534,
GCCSID=65534,
ENSCHEME=ASCII,  <--- WAS EBCDIC
DATE=ISO,
DATELEN=0,
DECARTH=DEC15,
DECIMAL=PERIOD,
DEFLANG=IBMCOB,
DELIM=DEFAULT,
MIXED=NO,
SQLDELI=DEFAULT,
DSQDELI=APOST,
SSID=DB2J,
STDSQL=NO,
TIME=ISO,
TIMELEN=0,
DYNRULS=YES,
LC_CTYPE=,
COMPAT=OFF

END

//*
//*******************************************************************************/
//*   LINK EDIT DSNHDECP.
//*   DSNHDECP IS A DATA-ONLY LOAD MODULE CONTAINING DEFAULT VALUES
//*   REQUIRED BY DB2 AND APPLICATION PROGRAMS.
//*   THIS STEP IS CREATED ONLY WHEN THE DEFAULTS SUPPLIED IN
//*   DSNHDECP ARE NOT SUITABLE.
//*******************************************************************************/
//DSNTIZQ EXEC PGM=IEWL,PARM='LIST,XREF,LET,RENT',
//              COND=(4,LT)
//ADSNLOAD DD  DISP=SHR,
//         DSN=DSN610.SDSNEXIT
//         DD  DISP=SHR,
//         DSN=DSN610.ADSNLOAD
//SYSPUNCH DD  DSN=&&LOADSET(DSNHDECA),DISP=(OLD,DELETE)
//SYSLMOD DD  DISP=SHR,
//         DSN=DB2V6100.SDSNEXIT
//SYSPRINT DD  SYSOUT=* 
//SYSUDUMP DD  SYSOUT=* 
//SYSUT1 DD  UNIT=SYSALLDA,SPACE=(1024,(50,50))
//SYSLIN DD  *
//     INCLUDE SYSPUNCH(DSNHDECA)
ORDER DSNAA
INCLUDE ADSNLOAD(DSNAA)
INCLUDE ADSNLOAD(DSNARIB)
Creating STOGROUPs

#include ADSNLOAD(DSNHDECP)
ENTRY DSNHDECP
MODE AMODE(24),RMODE(24)
NAME DSNHDECP(R)

Creating the Siebel databases

#include ADSNLOAD(DSNHDECP)
ENTRY DSNHDECP
MODE AMODE(24),RMODE(24)
NAME DSNHDECP(R)

//CREATEST JOB (999,POK), 'STOGROUP', NOTIFY=&SYSUID,
// CLASS=A, MSGCLASS=T, TIME=1439,
// MSGLEVEL=(1,1)
//***
//*** CREATE STOGROUPS FOR SIEBEL
//***
/*JOBPARM SYSAFF=SC04
//JOBLIB DD DSN=DB2V610D.RUNLIB.LOAD,DISP=SHR
// DD DSN=DSN610.SDSNLOAD,DISP=SHR
//CREATEST EXEC PGM=IKJEFT01,DYNAMNBR=20
//SYSPRINT DD SYSPRT=*’
//SYSTSPRT DD SYSPRINT=*’
//SYSSOUT DD SYSSPRINT=*’
//SYSUDUMP DD SYSUDUMP=*’
//SYSTSIN DD *
DSN S(DB2D)
RUN PROG(DSNTIAD) PLAN(DSNTIA61)
END
//SYSSIN DD *
CREATE STOGROUP STORP01
VOLUMES(SIEBE1,SIEBE2,SIEBE3,SIEBE4,SIEBE5)
VCAT DB2V610D;
GRANT USE OF STOGROUP STORP01 TO SSEEIM;
GRANT USE OF STOGROUP STORP01 TO SIEBEL WITH GRANT OPTION;
COMMIT;

Creating the Siebel databases

#include ADSNLOAD(DSNHDECP)
ENTRY DSNHDECP
MODE AMODE(24),RMODE(24)
NAME DSNHDECP(R)

//CREATEDB JOB (999,POK), 'CREATE DB', NOTIFY=&SYSUID,
// CLASS=A, MSGCLASS=T, TIME=1439,
// MSGLEVEL=(1,1)
//***
//*** CREATE SIEBEL DATABASES
Creating the Siebel table spaces

We used the following job to create the table spaces:

```
//CRELBL  JOB (999,POK), 'CREATE TBSPACE', NOTIFY=&SYSUID,
//     CLASS=A,MSGCLASS=T,TIME=1439,
//     MSGLEVEL=(1,1)
/*JOBPARM SYSAFF=SC04
//***
//***
//*** CREATE ALL THE TABLESPACES NEEDED FOR SIEBEL
//***
//***
//JOBLIB  DD DSN=DB2V610D.RUNLIB.LOAD,DISP=SHR
//        DD DSN=DSN610.SDSNLOAD,DISP=SHR
//CREATETB EXEC PGM=IKJEFT01,DYNAMNBR=20
//SYSPRINT DD SYSOUT=* 
//SYSTSPRT DD SYSOUT=* 
//SYSSIN DD *
DSN S(DB2D)
RUN PROG(DSNTIAD) PLAN(DSNTIA61)
END
//SYSIN DD *
CREATE DATABASE SIEBEL01 STOGROUP STOGRP01;
CREATE DATABASE SIEBEL02 STOGROUP STOGRP01;
CREATE DATABASE SIEBEL03 STOGROUP STOGRP01;
CREATE DATABASE SIEBEL04 STOGROUP STOGRP01;
CREATE DATABASE SIEBEL06 STOGROUP STOGRP01;
CREATE DATABASE SIEBEL07 STOGROUP STOGRP01;
CREATE DATABASE SIEBEL08 STOGROUP STOGRP01;
CREATE DATABASE SIEBEL09 STOGROUP STOGRP01;
CREATE DATABASE SIEBEL10 STOGROUP STOGRP01;
COMMIT;
//```
/SYSOUT DD SYSOUT=* 
/SYSUDUMP DD SYSOUT=* 
/SYSTSIN DD * 
  DSN S(DB2D) 
  RUN PROG(DSNTIAD) PLAN(DSNTIA61) 
  END 
/SYSIN DD *  

CREATE TABLESPACE FFFFO01 
  IN SIEBEL01 
  SEGSIZE 32 USING STOGROUP STOGRP01 
  BUFFERPOOL BP32K1 
  LOCKSIZE PAGE; COMMIT; 

CREATE TABLESPACE EEEEO01 
  IN SIEBEL02 
  SEGSIZE 32 USING STOGROUP STOGRP01 
  BUFFERPOOL BP16K1 
  LOCKSIZE PAGE; COMMIT; 

CREATE TABLESPACE GGGG010 
  IN SIEBEL03 
  SEGSIZE 32 USING STOGROUP STOGRP01 
  BUFFERPOOL BP16K1 
  LOCKSIZE PAGE; COMMIT; 

CREATE TABLESPACE GGGG011 
  IN SIEBEL03 
  SEGSIZE 32 USING STOGROUP STOGRP01 
  BUFFERPOOL BP16K1 
  LOCKSIZE PAGE; COMMIT; 

CREATE TABLESPACE GGGG012 
  IN SIEBEL03 
  SEGSIZE 32 USING STOGROUP STOGRP01 
  BUFFERPOOL BP16K1 
  LOCKSIZE PAGE; COMMIT; 

CREATE TABLESPACE GGGG013 
  IN SIEBEL03 
  SEGSIZE 32 USING STOGROUP STOGRP01 
  BUFFERPOOL BP16K1 
  LOCKSIZE PAGE; COMMIT; 

CREATE TABLESPACE GGGG014 
  IN SIEBEL03 
  SEGSIZE 32 USING STOGROUP STOGRP01 
  BUFFERPOOL BP16K1 
  LOCKSIZE PAGE; COMMIT;
CREATE TABLESPACE GGGGG015
  IN SIEBEL03
  SEGSIZE 32 USING STOGROUP STOGRP01
  BUFFERPOOL  BP16K1
  LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE GGGGG016
  IN SIEBEL03
  SEGSIZE 32 USING STOGROUP STOGRP01
  BUFFERPOOL  BP16K1
  LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE GGGGG017
  IN SIEBEL03
  SEGSIZE 32 USING STOGROUP STOGRP01
  BUFFERPOOL  BP16K1
  LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE GGGGG001
  IN SIEBEL04
  SEGSIZE 32 USING STOGROUP STOGRP01
  BUFFERPOOL  BP16K1
  LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE GGGGG002
  IN SIEBEL04
  SEGSIZE 32 USING STOGROUP STOGRP01
  BUFFERPOOL  BP16K1
  LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE GGGGG003
  IN SIEBEL04
  SEGSIZE 32 USING STOGROUP STOGRP01
  BUFFERPOOL  BP16K1
  LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE GGGGG004
  IN SIEBEL04
  SEGSIZE 32 USING STOGROUP STOGRP01
  BUFFERPOOL  BP16K1
  LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE GGGGG005
  IN SIEBEL04
  SEGSIZE 32 USING STOGROUP STOGRP01
  BUFFERPOOL  BP16K1
  LOCKSIZE PAGE; COMMIT;
CREATE TABLESPACE GGGGG006
IN SIEBEL04
SEGSIZE 32 USING STOGROUP STORP01
BUFFERPOOL BP16K1
LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE GGGGG007
IN SIEBEL04
SEGSIZE 32 USING STOGROUP STORP01
BUFFERPOOL BP16K1
LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE GGGGG008
IN SIEBEL04
SEGSIZE 32 USING STOGROUP STORP01
BUFFERPOOL BP16K1
LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE HHHHH001
IN SIEBEL05
SEGSIZE 32 USING STOGROUP STORP01
PRIQTY 7200 SECQTY 720
BUFFERPOOL BP16K1
LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE HHHHH002
IN SIEBEL05
SEGSIZE 32 USING STOGROUP STORP01
PRIQTY 7200 SECQTY 720
BUFFERPOOL BP16K1
LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE BBBBB024
IN SIEBEL06
SEGSIZE 32 USING STOGROUP STORP01
BUFFERPOOL BP1
LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE BBBBB025
IN SIEBEL06
SEGSIZE 32 USING STOGROUP STORP01
BUFFERPOOL BP1
LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE BBBBB026
IN SIEBEL06
SEGSIZE 32 USING STOGROUP STORP01
BUFFERPOOL BP1
LOCKSIZE PAGE; COMMIT;
CREATE TABLESPACE BBBBB027
  IN SIEBEL06
  SEGSIZE 32 USING STOGROUP STOGRP01
  BUFFERPOOL BP1
  LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE BBBBB028
  IN SIEBEL06
  SEGSIZE 32 USING STOGROUP STOGRP01
  BUFFERPOOL BP1
  LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE BBBBB029
  IN SIEBEL06
  SEGSIZE 32 USING STOGROUP STOGRP01
  BUFFERPOOL BP1
  LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE BBBBB031
  IN SIEBEL06
  SEGSIZE 32 USING STOGROUP STOGRP01
  BUFFERPOOL BP1
  LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE BBBBB032
  IN SIEBEL06
  SEGSIZE 32 USING STOGROUP STOGRP01
  BUFFERPOOL BP1
  LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE BBBBB033
  IN SIEBEL06
  SEGSIZE 32 USING STOGROUP STOGRP01
  BUFFERPOOL BP1
  LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE BBBBB034
  IN SIEBEL06
  SEGSIZE 32 USING STOGROUP STOGRP01
  BUFFERPOOL BP1
  LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE BBBBB035
  IN SIEBEL06
  SEGSIZE 32 USING STOGROUP STOGRP01
  BUFFERPOOL BP1
  LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE BBBBB036
IN SIEBEL06
SEG SIZE 32 USING STOGROUP  STOGRP01
BUFFERPOOL  BP1
LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE BBBBB037
IN SIEBEL06
SEG SIZE 32 USING STOGROUP  STOGRP01
BUFFERPOOL  BP1
LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE BBBBB038
IN SIEBEL06
SEG SIZE 32 USING STOGROUP  STOGRP01
BUFFERPOOL  BP1
LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE BBBBB039
IN SIEBEL06
SEG SIZE 32 USING STOGROUP  STOGRP01
BUFFERPOOL  BP1
LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE BBBBB040
IN SIEBEL06
SEG SIZE 32 USING STOGROUP  STOGRP01
BUFFERPOOL  BP1
LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE BBBBB041
IN SIEBEL06
SEG SIZE 32 USING STOGROUP  STOGRP01
BUFFERPOOL  BP1
LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE BBBBB042
IN SIEBEL06
SEG SIZE 32 USING STOGROUP  STOGRP01
BUFFERPOOL  BP1
LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE BBBBB043
IN SIEBEL06
SEG SIZE 32 USING STOGROUP  STOGRP01
BUFFERPOOL  BP1
LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE BBBBB044
IN SIEBEL06
SEGSIZE 32 USING STOGROUP STOGP01
BUFFERPOOL BP1
LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE CCCCC002 IN SIEBEL06
SEGSIZE 32 USING STOGROUP STOGP01
BUFFERPOOL BP16K1
LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE DDDDD015 IN SIEBEL06
SEGSIZE 32 USING STOGROUP STOGP01
BUFFERPOOL BP1
LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE DDDDD016 IN SIEBEL06
SEGSIZE 32 USING STOGROUP STOGP01
BUFFERPOOL BP1
LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE DDDDD017 IN SIEBEL06
SEGSIZE 32 USING STOGROUP STOGP01
BUFFERPOOL BP1
LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE DDDDD018 IN SIEBEL06
SEGSIZE 32 USING STOGROUP STOGP01
BUFFERPOOL BP1
LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE DDDDD019 IN SIEBEL06
SEGSIZE 32 USING STOGROUP STOGP01
BUFFERPOOL BP1
LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE DDDDD020 IN SIEBEL06
SEGSIZE 32 USING STOGROUP STOGP01
BUFFERPOOL BP1
LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE DDDDD021 IN SIEBEL06
SEGSIZE 32 USING STOGROUP STOGP01
BUFFERPOOL  BP1
LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE DDDDD022
  IN SIEBEL06
  SEGSIZE 32 USING STOGROUP STOGRP01
  BUFFERPOOL  BP1
  LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE DDDDD023
  IN SIEBEL06
  SEGSIZE 32 USING STOGROUP STOGRP01
  BUFFERPOOL  BP1
  LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE DDDDD024
  IN SIEBEL06
  SEGSIZE 32 USING STOGROUP STOGRP01
  BUFFERPOOL  BP1
  LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE DDDDD055
  IN SIEBEL06
  SEGSIZE 32 USING STOGROUP STOGRP01
  BUFFERPOOL  BP1
  LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE AAAAA001
  IN SIEBEL07
  SEGSIZE 32 USING STOGROUP STOGRP01
  BUFFERPOOL  BP16K1
  LOCKSIZE ROW; COMMIT;

CREATE TABLESPACE AAAAA002
  IN SIEBEL07
  SEGSIZE 32 USING STOGROUP STOGRP01
  BUFFERPOOL  BP1
  LOCKSIZE ROW; COMMIT;

CREATE TABLESPACE AAAAA003
  IN SIEBEL07
  SEGSIZE 32 USING STOGROUP STOGRP01
  BUFFERPOOL  BP1
  LOCKSIZE ROW; COMMIT;

CREATE TABLESPACE AAAAA004
  IN SIEBEL07
  SEGSIZE 32 USING STOGROUP STOGRP01
  BUFFERPOOL  BP1
LOCKSIZE ROW; COMMIT;

CREATE TABLESPACE AAAAA005  
IN SIEBEL07  
SEGSIZE 32 USING STOGROUP STOGRP01  
BUFFERPOOL BP1  
LOCKSIZE ROW; COMMIT;

CREATE TABLESPACE BBBBB045  
IN SIEBEL07  
SEGSIZE 32 USING STOGROUP STOGRP01  
BUFFERPOOL BP1  
LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE BBBBB046  
IN SIEBEL07  
SEGSIZE 32 USING STOGROUP STOGRP01  
BUFFERPOOL BP1  
LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE BBBBB047  
IN SIEBEL07  
SEGSIZE 32 USING STOGROUP STOGRP01  
BUFFERPOOL BP1  
LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE BBBBB048  
IN SIEBEL07  
SEGSIZE 32 USING STOGROUP STOGRP01  
BUFFERPOOL BP1  
LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE BBBBB049  
IN SIEBEL07  
SEGSIZE 32 USING STOGROUP STOGRP01  
BUFFERPOOL BP1  
LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE BBBBB050  
IN SIEBEL07  
SEGSIZE 32 USING STOGROUP STOGRP01  
BUFFERPOOL BP1  
LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE BBBBB051  
IN SIEBEL07  
SEGSIZE 32 USING STOGROUP STOGRP01  
BUFFERPOOL BP1  
LOCKSIZE PAGE; COMMIT;
<table>
<thead>
<tr>
<th>Tablespace Name</th>
<th>Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>BBBBB052</td>
<td>IN SIEBEL07 SEGSIZE 32 USING STOGROUP STOGRP01</td>
</tr>
<tr>
<td>BBBBB053</td>
<td>IN SIEBEL07 SEGSIZE 32 USING STOGROUP STOGRP01</td>
</tr>
<tr>
<td>BBBBB054</td>
<td>IN SIEBEL07 SEGSIZE 32 USING STOGROUP STOGRP01</td>
</tr>
<tr>
<td>BBBBB055</td>
<td>IN SIEBEL07 SEGSIZE 32 USING STOGROUP STOGRP01</td>
</tr>
<tr>
<td>BBBBB056</td>
<td>IN SIEBEL07 SEGSIZE 32 USING STOGROUP STOGRP01</td>
</tr>
<tr>
<td>BBBBB057</td>
<td>IN SIEBEL07 SEGSIZE 32 USING STOGROUP STOGRP01</td>
</tr>
<tr>
<td>BBBBB058</td>
<td>IN SIEBEL07 SEGSIZE 32 USING STOGROUP STOGRP01</td>
</tr>
<tr>
<td>BBBBB059</td>
<td>IN SIEBEL07 SEGSIZE 32 USING STOGROUP STOGRP01</td>
</tr>
</tbody>
</table>

CREATE TABLESPACE BBBBB052
IN SIEBEL07
SEGSIZE 32 USING STOGROUP STOGRP01
BUFFERPOOL BP1
LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE BBBBB053
IN SIEBEL07
SEGSIZE 32 USING STOGROUP STOGRP01
BUFFERPOOL BP1
LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE BBBBB054
IN SIEBEL07
SEGSIZE 32 USING STOGROUP STOGRP01
BUFFERPOOL BP1
LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE BBBBB055
IN SIEBEL07
SEGSIZE 32 USING STOGROUP STOGRP01
BUFFERPOOL BP1
LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE BBBBB056
IN SIEBEL07
SEGSIZE 32 USING STOGROUP STOGRP01
BUFFERPOOL BP1
LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE BBBBB057
IN SIEBEL07
SEGSIZE 32 USING STOGROUP STOGRP01
BUFFERPOOL BP1
LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE BBBBB058
IN SIEBEL07
SEGSIZE 32 USING STOGROUP STOGRP01
BUFFERPOOL BP1
LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE BBBBB059
IN SIEBEL07
SEGSIZE 32 USING STOGROUP STOGRP01
BUFFERPOOL BP1
LOCKSIZE PAGE; COMMIT;
CREATE TABLESPACE BBBBB060
   IN SIEBEL07
   SEGSIZE 32 USING STOGROUP STOGRP01
   BUFFERPOOL BP1
   LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE DDDDD005
   IN SIEBEL07
   SEGSIZE 32 USING STOGROUP STOGRP01
   BUFFERPOOL BP1
   LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE DDDDD006
   IN SIEBEL07
   SEGSIZE 32 USING STOGROUP STOGRP01
   BUFFERPOOL BP1
   LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE DDDDD007
   IN SIEBEL07
   SEGSIZE 32 USING STOGROUP STOGRP01
   BUFFERPOOL BP1
   LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE DDDDD008
   IN SIEBEL07
   SEGSIZE 32 USING STOGROUP STOGRP01
   BUFFERPOOL BP1
   LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE DDDDD009
   IN SIEBEL07
   SEGSIZE 32 USING STOGROUP STOGRP01
   BUFFERPOOL BP1
   LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE DDDDD010
   IN SIEBEL07
   SEGSIZE 32 USING STOGROUP STOGRP01
   BUFFERPOOL BP1
   LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE DDDDD011
   IN SIEBEL07
   SEGSIZE 32 USING STOGROUP STOGRP01
   BUFFERPOOL BP1
   LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE DDDDD012
IN SIEBEL07
SEG SIZE 32 USING STOGROUP STORP01
BUFFERPOOL BP1
LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE DDDDDO13
IN SIEBEL07
SEG SIZE 32 USING STOGROUP STORP01
BUFFERPOOL BP1
LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE DDDDDO14
IN SIEBEL07
SEG SIZE 32 USING STOGROUP STORP01
BUFFERPOOL BP1
LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE BBBBB001
IN SIEBEL08
SEG SIZE 32 USING STOGROUP STORP01
BUFFERPOOL BP1
LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE BBBBB002
IN SIEBEL08
SEG SIZE 32 USING STOGROUP STORP01
BUFFERPOOL BP1
LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE BBBBB003
IN SIEBEL08
SEG SIZE 32 USING STOGROUP STORP01
BUFFERPOOL BP1
LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE BBBBB004
IN SIEBEL08
SEG SIZE 32 USING STOGROUP STORP01
BUFFERPOOL BP1
LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE BBBBB061
IN SIEBEL08
SEG SIZE 32 USING STOGROUP STORP01
BUFFERPOOL BP1
LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE BBBBB062
IN SIEBEL08
SEGSIZE 32 USING STOGROUP      STOGRP01
BUFFERPOOL   BP1
LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE CCCCC001
    IN SIEBEL08
    SEGSIZE 32 USING STOGROUP      STOGRP01
    BUFFERPOOL   BP1
    LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE DDDDD001
    IN SIEBEL08
    SEGSIZE 32 USING STOGROUP      STOGRP01
    BUFFERPOOL   BP1
    LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE DDDDD002
    IN SIEBEL08
    SEGSIZE 32 USING STOGROUP      STOGRP01
    BUFFERPOOL   BP1
    LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE DDDDD003
    IN SIEBEL08
    SEGSIZE 32 USING STOGROUP      STOGRP01
    BUFFERPOOL   BP1
    LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE DDDDD041
    IN SIEBEL08
    SEGSIZE 32 USING STOGROUP      STOGRP01
    BUFFERPOOL   BP1
    LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE DDDDD042
    IN SIEBEL08
    SEGSIZE 32 USING STOGROUP      STOGRP01
    BUFFERPOOL   BP1
    LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE DDDDD043
    IN SIEBEL08
    SEGSIZE 32 USING STOGROUP      STOGRP01
    BUFFERPOOL   BP1
    LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE DDDDD044
    IN SIEBEL08
    SEGSIZE 32 USING STOGROUP      STOGRP01
BUFFERPOOL  BP1
LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE DDDDD045
IN SIEBEL08
SEGSIZE 32 USING STOGROUP STOGRP01
BUFFERPOOL  BP1
LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE DDDDD046
IN SIEBEL08
SEGSIZE 32 USING STOGROUP STOGRP01
BUFFERPOOL  BP1
LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE DDDDD047
IN SIEBEL08
SEGSIZE 32 USING STOGROUP STOGRP01
BUFFERPOOL  BP1
LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE DDDDD048
IN SIEBEL08
SEGSIZE 32 USING STOGROUP STOGRP01
BUFFERPOOL  BP1
LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE DDDDD049
IN SIEBEL08
SEGSIZE 32 USING STOGROUP STOGRP01
BUFFERPOOL  BP1
LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE DDDDD050
IN SIEBEL08
SEGSIZE 32 USING STOGROUP STOGRP01
BUFFERPOOL  BP1
LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE DDDDD051
IN SIEBEL08
SEGSIZE 32 USING STOGROUP STOGRP01
BUFFERPOOL  BP1
LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE DDDDD052
IN SIEBEL08
SEGSIZE 32 USING STOGROUP STOGRP01
BUFFERPOOL  BP1
LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE DDDDD053
IN SIEBEL08
SEGSIZE 32 USING STOGROUP STORGP01
BUFFERPOOL BP1
LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE DDDDD054
IN SIEBEL08
SEGSIZE 32 USING STOGROUP STORGP01
BUFFERPOOL BP1
LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE BBBBB006
IN SIEBEL09
SEGSIZE 32 USING STOGROUP STORGP01
BUFFERPOOL BP1
LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE BBBBB007
IN SIEBEL09
SEGSIZE 32 USING STOGROUP STORGP01
BUFFERPOOL BP1
LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE BBBBB008
IN SIEBEL09
SEGSIZE 32 USING STOGROUP STORGP01
BUFFERPOOL BP1
LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE BBBBB009
IN SIEBEL09
SEGSIZE 32 USING STOGROUP STORGP01
BUFFERPOOL BP1
LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE BBBBB010
IN SIEBEL09
SEGSIZE 32 USING STOGROUP STORGP01
BUFFERPOOL BP1
LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE BBBBB011
IN SIEBEL09
SEGSIZE 32 USING STOGROUP STORGP01
BUFFERPOOL BP1
LOCKSIZE PAGE; COMMIT;
CREATE TABLESPACE BBBBB012
IN SIEBEL09
SEGSIZE 32 USING STOGROUP STOGRP01
BUFFERPOOL BP1
LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE BBBBB013
IN SIEBEL09
SEGSIZE 32 USING STOGROUP STOGRP01
BUFFERPOOL BP1
LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE BBBBB014
IN SIEBEL09
SEGSIZE 32 USING STOGROUP STOGRP01
BUFFERPOOL BP1
LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE BBBBB015
IN SIEBEL09
SEGSIZE 32 USING STOGROUP STOGRP01
BUFFERPOOL BP1
LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE BBBBB016
IN SIEBEL09
SEGSIZE 32 USING STOGROUP STOGRP01
BUFFERPOOL BP1
LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE DDDDD030
IN SIEBEL09
SEGSIZE 32 USING STOGROUP STOGRP01
BUFFERPOOL BP1
LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE DDDDD031
IN SIEBEL09
SEGSIZE 32 USING STOGROUP STOGRP01
BUFFERPOOL BP1
LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE DDDDD032
IN SIEBEL09
SEGSIZE 32 USING STOGROUP STOGRP01
BUFFERPOOL BP1
LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE DDDDD033
IN SIEBEL09
SEGSIZE 32 USING STOGROUP STOGRP01
BUFFERPOOL BP1
LOCKSIZE PAGE; COMMIT;
IN SIEBEL09
SEG SIZE 32 USING STOGROUP STORP01
BUFFERPOOL BP1
LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE DDDDD034
IN SIEBEL09
SEG SIZE 32 USING STOGROUP STORP01
BUFFERPOOL BP1
LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE DDDDD035
IN SIEBEL09
SEG SIZE 32 USING STOGROUP STORP01
BUFFERPOOL BP1
LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE DDDDD036
IN SIEBEL09
SEG SIZE 32 USING STOGROUP STORP01
BUFFERPOOL BP1
LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE DDDDD037
IN SIEBEL09
SEG SIZE 32 USING STOGROUP STORP01
BUFFERPOOL BP1
LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE DDDDD038
IN SIEBEL09
SEG SIZE 32 USING STOGROUP STORP01
BUFFERPOOL BP1
LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE DDDDD039
IN SIEBEL09
SEG SIZE 32 USING STOGROUP STORP01
BUFFERPOOL BP1
LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE DDDDD040
IN SIEBEL09
SEG SIZE 32 USING STOGROUP STORP01
BUFFERPOOL BP1
LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE AAAAA006
IN SIEBEL10
CREATE TABLESPACE AAAAA007
  IN SIEBEL10
  SEGSIZE 32 USING STOGROUP  STOGRP01
  BUFFERPOOL   BP1
  LOCKSIZE ROW; COMMIT;

CREATE TABLESPACE AAAAA008
  IN SIEBEL10
  SEGSIZE 32 USING STOGROUP  STOGRP01
  BUFFERPOOL   BP1
  LOCKSIZE ROW; COMMIT;

CREATE TABLESPACE BBBBB017
  IN SIEBEL10
  SEGSIZE 32 USING STOGROUP  STOGRP01
  BUFFERPOOL   BP1
  LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE BBBBB018
  IN SIEBEL10
  SEGSIZE 32 USING STOGROUP  STOGRP01
  BUFFERPOOL   BP1
  LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE BBBBB019
  IN SIEBEL10
  SEGSIZE 32 USING STOGROUP  STOGRP01
  BUFFERPOOL   BP1
  LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE BBBBB020
  IN SIEBEL10
  SEGSIZE 32 USING STOGROUP  STOGRP01
  BUFFERPOOL   BP1
  LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE BBBBB021
  IN SIEBEL10
  SEGSIZE 32 USING STOGROUP  STOGRP01
  BUFFERPOOL   BP1
  LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE BBBBB022
  IN SIEBEL10
  SEGSIZE 32 USING STOGROUP  STOGRP01
Creating Siebel tables and indexes

Here is a subset of the job we used to create Siebel tables and indexes.

```sql
BUFFERPOOL   BP1
LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE BBBBB023
  IN SIEBEL10
  SEGSIZE 32 USING STOGROUP     STOGRP01
  BUFFERPOOL   BP1
  LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE DDDDD025
  IN SIEBEL10
  SEGSIZE 32 USING STOGROUP     STOGRP01
  BUFFERPOOL   BP1
  LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE DDDDD026
  IN SIEBEL10
  SEGSIZE 32 USING STOGROUP     STOGRP01
  BUFFERPOOL   BP1
  LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE DDDDD027
  IN SIEBEL10
  SEGSIZE 32 USING STOGROUP     STOGRP01
  BUFFERPOOL   BP1
  LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE DDDDD028
  IN SIEBEL10
  SEGSIZE 32 USING STOGROUP     STOGRP01
  BUFFERPOOL   BP1
  LOCKSIZE PAGE; COMMIT;

CREATE TABLESPACE DDDDD029
  IN SIEBEL10
  SEGSIZE 32 USING STOGROUP     STOGRP01
  BUFFERPOOL   BP1
  LOCKSIZE PAGE; COMMIT;
```
//***
//*** CREATE ALL THE TABLE AND INDEXES NEEDED FOR SIEBEL
//***

//JOBLIB DD DSN=DB2V610D.RUNLIB.LOAD,DISP=SHR
// DD DSN=DSN610.SDSNLOAD,DISP=SHR
//CREATETB EXEC PGM=IKJEFT01,DYNAMNBR=20
//SYSPRINT DD SYSOUT=*  
//SYSSTSPRT DD SYSOUT=*  
//SYSOUT DD SYSOUT=*  
//SYSDUMP DD SYSOUT=*  
//SYSTIN DD *
    DSN S(DB2D)
    RUN PROG(DSNTIAD) PLAN(DSNTIA61)
END

//SYSSIN DD *
    CREATE TABLE EIM_ACCNT_DTL ( 
    MS_IDENT NUMERIC(15,0) GENERATED ALWAYS AS IDENTITY, 
    ROW_ID VARCHAR(15) NOT NULL, 
    CREATED TIMESTAMP DEFAULT, 
    CREATED_BY VARCHAR(15), 
    LAST_UPD TIMESTAMP DEFAULT, 
    LAST_UPD_BY VARCHAR(15), 
    MODIFICATION_NUM NUMERIC(10,0) DEFAULT 0, 
    CONFLICT_ID VARCHAR(15) DEFAULT '0', 
    IF_ROW_BATCH_NUM NUMERIC(15,0) DEFAULT 0 NOT NULL, 
    IF_ROW_STAT VARCHAR(30) NOT NULL, 
    NAME VARCHAR(100) NOT NULL, 
    ACCNT_BI VARCHAR(15), 
    ACCNT_BU VARCHAR(50), 
    ACC_PR_LOGO CHAR(1), 
    ASGN_IT_TYPE_NAME VARCHAR(75), 
    ATT_COMMENTS VARCHAR(250), 
    ATT_FILE_EXT VARCHAR(10), 
    ATT_FILE_NAME VARCHAR(200), 
    ATT_FILE_SRC_TYPE VARCHAR(30), 
    COMMENTS VARCHAR(250), 
    EXT_ATTRIB_01 VARCHAR(100), 
    EXT_ATTRIB_02 VARCHAR(100), 
    EXT_ATTRIB_03 VARCHAR(30), 
    EXT_ATTRIB_04 VARCHAR(30), 
    EXT_ATTRIB_05 VARCHAR(30), 
    EXT_ATTRIB_06 VARCHAR(30), 
    EXT_ATTRIB_07 VARCHAR(30), 
    EXT_ATTRIB_08 CHAR(1), 
    EXT_ATTRIB_09 CHAR(1), 
    EXT_ATTRIB_10 CHAR(1), 
    EXT_ATTRIB_11 CHAR(1),

154 Migrating Siebel Database from DB2/Oracle for NT to DB2 for OS/390
EXT_ATTRIB_12                  TIMESTAMP,
EXT_ATTRIB_13                  TIMESTAMP,
EXT_ATTRIB_14                  NUMERIC(22,7),
EXT_ATTRIB_15                  NUMERIC(22,7),
EXT_ATTRIB_16                  NUMERIC(22,7),
EXT_ATTRIB_17                  NUMERIC(22,7),
EXT_ATTRIB_18                  NUMERIC(22,7),
EXT_ATTRIB_19                  NUMERIC(22,7),
EXT_ATTRIB_20                  NUMERIC(22,7),
EXT_ATTRIB_21                  NUMERIC(22,7),
EXT_ATTRIB_22                  NUMERIC(22,7),
EXT_ATTRIB_23                  NUMERIC(22,7),
EXT_ATTRIB_24                  NUMERIC(22,7),
EXT_ATTRIB_25                  NUMERIC(22,7),
EXT_ATTRIB_26                  TIMESTAMP,
EXT_ATTRIB_27                  TIMESTAMP,
EXT_ATTRIB_28                  TIMESTAMP,
EXT_ATTRIB_29                  TIMESTAMP,
EXT_ATTRIB_30                  TIMESTAMP,
EXT_ATTRIB_31                  TIMESTAMP,
EXT_ATTRIB_32                  TIMESTAMP,
EXT_ATTRIB_33                  TIMESTAMP,
EXT_ATTRIB_34                  VARCHAR(50),
EXT_ATTRIB_35                  VARCHAR(50),
EXT_ATTRIB_36                  VARCHAR(50),
EXT_ATTRIB_37                  VARCHAR(50),
EXT_ATTRIB_38                  VARCHAR(50),
EXT_ATTRIB_39                  VARCHAR(50),
EXT_ATTRIB_40                  VARCHAR(50),
EXT_ATTRIB_41                  VARCHAR(50),
EXT_ATTRIB_42                  VARCHAR(50),
EXT_ATTRIB_43                  VARCHAR(50),
EXT_ATTRIB_44                  VARCHAR(100),
EXT_ATTRIB_45                  VARCHAR(100),
EXT_ATTRIB_46                  VARCHAR(100),
EXT_ATTRIB_47                  VARCHAR(255),
IF_ROW_MERGE_ID                VARCHAR(15),
IF_ROW_STAT_NUM                NUMERIC(10,0),
INCL_EXCL_CD                   VARCHAR(30),
IT_COMMENTS                    VARCHAR(250),
IT_EXPERTISE_CD                VARCHAR(30),
IT_LO_CHAR1                    VARCHAR(100),
IT_LO_CHAR2                    VARCHAR(50),
IT_LO_CHAR3                    VARCHAR(50),
IT_LO_CHAR4                    VARCHAR(250),
IT_LO_NUM1                     NUMERIC(10,0),
IT_LO_NUM2                     NUMERIC(10,0),
IT_LO_NUM3                     NUMERIC(10,0),
IT_LO_NUM4                     NUMERIC(10,0),
LOC            VARCHAR(50),
NOTE_NOTE      LONG VARCHAR,
NOTE_NOTE_TYPE VARCHAR(30),
NOTE_PRIV_FLG  CHAR(1),
ORG_SKL_NAME   VARCHAR(50),
T_ACNT_ATT_PARROW VARCHAR(15),
T_ACNT_ATT__EXS CHAR(1),
T_ACNT_ATT__RID VARCHAR(15),
T_ACNT_ATT__STA NUMERIC(10,0),
T_ACNT_ATT__UNQ CHAR(1),
T_DELETED_ROW_ID VARCHAR(15),
T_EXPORTED_ROW_ID VARCHAR(15),
T_MERGED_ROW_ID VARCHAR(15),
T_NOTEACCNT_SRCROW VARCHAR(15),
T_NOTEACCNT__EXS CHAR(1),
T_NOTEACCNT__RID VARCHAR(15),
T_NOTEACCNT__STA NUMERIC(10,0),
T_NOTEACCNT__UNQ CHAR(1),
T_ORGEXT_XM_PARROW VARCHAR(15),
T_ORGEXT_XM__EXS CHAR(1),
T_ORGEXT_XM__RID VARCHAR(15),
T_ORGEXT_XM__STA NUMERIC(10,0),
T_ORGEXT_XM__UNQ CHAR(1),
T_ORGSKILLI ORGSKI VARCHAR(15),
T_ORGSKILLI__EXS CHAR(1),
T_ORGSKILLI__RID VARCHAR(15),
T_ORGSKILLI__STA NUMERIC(10,0),
T_ORGSKILLI__UNQ CHAR(1),
T_ORG_EXT_BU_ID VARCHAR(15),
T_ORG_EXT_X__EXS CHAR(1),
T_ORG_EXT_X__RID VARCHAR(15),
T_ORG_EXT_X__STA NUMERIC(10,0),
T_ORG_EXT_X__UNQ CHAR(1),
T_ORG_EXT__EXS CHAR(1),
T_ORG_EXT__RID VARCHAR(15),
T_ORG_EXT__STA NUMERIC(10,0),
T_ORG_EXT__UNQ CHAR(1),
T_ORG_SKILL__ORG_ID VARCHAR(15),
T_ORG_SKILL__EXS CHAR(1),
T_ORG_SKILL__RID VARCHAR(15),
T_ORG_SKILL__STA NUMERIC(10,0),
T_ORG_SKILL__UNQ CHAR(1),
XM_NAME        VARCHAR(100),
XM_TYPE        VARCHAR(30))
IN SIEBEL04.GGGG001
;
COMMIT;
GRANT SELECT, INSERT, UPDATE, DELETE ON EIM_ACCNT_DTL TO SSEROLE
;
COMMIT;
CREATE UNIQUE INDEX EIM_ACCNT_DTL_U1 ON EIM_ACCNT_DTL
(ROW_ID, IF_ROW_BATCH_NUM)
CLUSTER USING STOGROUP STOGRP01 PRIQTY 720 SECQTY 720 DEFINE NO CLOSE YES;
COMMIT;
CREATE INDEX EIM_ACCNT_DTL_T01 ON EIM_ACCNT_DTL
  (IF_ROW_BATCH_NUM, T_DELETED_ROW_ID)
USING STOGROUP STOGRP01 PRIQTY 720 SECQTY 720 DEFINE NO CLOSE YES;
COMMIT;
CREATE INDEX EIM_ACCNT_DTL_T02 ON EIM_ACCNT_DTL
  (IF_ROW_BATCH_NUM, T_ORG_EXT__RID)
USING STOGROUP STOGRP01 PRIQTY 720 SECQTY 720 DEFINE NO CLOSE YES;
COMMIT;
CREATE INDEX EIM_ACCNT_DTL_T03 ON EIM_ACCNT_DTL
  (IF_ROW_BATCH_NUM, T_ACCNT_ATT__RID)
USING STOGROUP STOGRP01 PRIQTY 720 SECQTY 720 DEFINE NO CLOSE YES;
COMMIT;
CREATE INDEX EIM_ACCNT_DTL_T04 ON EIM_ACCNT_DTL
  (IF_ROW_BATCH_NUM, NAME, LOC, T_ORG_EXT_BU_ID)
USING STOGROUP STOGRP01 PRIQTY 720 SECQTY 720 DEFINE NO CLOSE YES;
COMMIT;
CREATE INDEX EIM_ACCNT_DTL_T05 ON EIM_ACCNT_DTL
  (IF_ROW_BATCH_NUM, T_ACCNT_ATT_PARROW, ATT_FILE_NAME, ATT_FILE_EXT)
USING STOGROUP STOGRP01 PRIQTY 720 SECQTY 720 DEFINE NO CLOSE YES;
COMMIT;
CREATE INDEX EIM_ACCNT_DTL_T06 ON EIM_ACCNT_DTL
  (IF_ROW_BATCH_NUM, T_NOTEACCNT__RID)
USING STOGROUP STOGRP01 PRIQTY 720 SECQTY 720 DEFINE NO CLOSE YES;
COMMIT;
CREATE INDEX EIM_ACCNT_DTL_T07 ON EIM_ACCNT_DTL
  (IF_ROW_BATCH_NUM, T_ORG_EXT_X__RID)
USING STOGROUP STOGRP01 PRIQTY 720 SECQTY 720 DEFINE NO CLOSE YES;
COMMIT;
CREATE INDEX EIM_ACCNT_DTL_T08 ON EIM_ACCNT_DTL
  (IF_ROW_BATCH_NUM, T_ORGEXT_XM__RID)
USING STOGROUP STOGRP01 PRIQTY 720 SECQTY 720 DEFINE NO CLOSE YES;
COMMIT;
CREATE INDEX EIM_ACCNT_DTL_T09 ON EIM_ACCNT_DTL
  (IF_ROW_BATCH_NUM, T_ORGEXT_XM_PARROW, XM_TYPE, XM_NAME)
USING STOGROUP STOGRP01 PRIQTY 720 SECQTY 720 DEFINE NO CLOSE YES;
COMMIT;
CREATE INDEX EIM_ACCNT_DTL_T10 ON EIM_ACCNT_DTL
  (IF_ROW_BATCH_NUM, T_ORG_SKILL__RID)
USING STOGROUP STOGRP01 PRIQTY 720 SECQTY 720 DEFINE NO CLOSE YES;
COMMIT;
CREATE INDEX EIM_ACCNT_DTL_T11 ON EIM_ACCNT_DTL
  (IF_ROW_BATCH_NUM, T_ORG_SKILL_ORG_ID, ORG_SKL_NAME)
USING STOGROUP STOGRP01 PRIQTY 720 SECQTY 720 DEFINE NO CLOSE YES;
COMMIT;
CREATE INDEX EIM_ACCNT_DTL_T12 ON EIM_ACCNT_DTL
Creating stored procedure objects

Here is the job we used to create the temporary database and table spaces required to enable the stored procedures for the Siebel application:

```sql
//CREATETP JOB (999,POK), 'CREATE TP', NOTIFY=&SYSUID, 00010023
//   CLASS=A, MSGCLASS=T, TIME=1439,
//   MSGLEVEL=(1,1) 00030017
//***
/*JOBPARM SYSAFF=SC04
//JOBLIB DD DSN=DB2V610D.RUNLIB.LOAD, DISP=SHR
// DD DSN=DSN610.SDSNLOAD, DISP=SHR
//CREATEDB EXEC PGM=IKJEFT01, DYNAMNBR=20
//SYSPRINT DD SYSOUT=* //SYSTSPRT DD SYSOUT=* //SYSDUMP DD SYSOUT=* //SYSTSIN DD *
// DSN S(DB2D)
RIN PROG(DSNTIAD) PLAN(DSNTIA61)
END
//SYSIN DD *
CREATE DATABASE SIEBELTP AS TEMP STOGROUP STOGRP01;
CREATE TABLESPACE TMPSPAC1
   IN SIEBELTP
   USING STOGROUP STOGRP01 PRIQTY 720 SECQTY 720 BUFFERPOOL BP7;
CREATE TABLESPACE TMPSPAC2
   IN SIEBELTP
   USING STOGROUP STOGRP01 PRIQTY 720 SECQTY 720 BUFFERPOOL BP16K11;
CREATE TABLESPACE TMPSPAC3
   IN SIEBELTP
   USING STOGROUP STOGRP01 PRIQTY 720 SECQTY 720 BUFFERPOOL BP32K1;
COMMIT;
/*
//
```

Migrating Siebel Database from DB2/Oracle for NT to DB2 for OS/390
Appendix E. DB2 DSNTIAUL and LOAD jobs

This appendix shows the DB2 for OS/390 DSNTIAUL and LOAD process.

A sample JOB follows, which creates a remote package for DSNTIAUL and a plan which contains the package list for the remote package. It then executes the DSNTIAUL utility to unload the data from the DB2 for NT server, edits the control dataset to change the target database names, then loads the data into the new tables. Note that you cannot use the PARMS('SQL') option with remote access.

Each table unloaded requires its own SYSRECxx DD statement number sequentially starting with xx between 00 - 99. Because of the volume of data and the fact that DSNTIAUL unloads varchar columns as fixed-length columns, there can be a lot of blank space in the load data sets. It is recommended that DFSMS-managed storage groups be used with the compression option.
//DSNTIAUL JOB TIAUL,MSGCLASS=H,CLASS=A,MSGLEVEL=(1,1),
//      REGION=5M,
//      USER=USRTO01, PASSWORD=USRTO01
//********************************************************************
// * NAME = BIND DSNTIAUL REMOTE
// * DESCRIPTIVE NAME = BIND DSNTIAUL ON THE REMOTE SERVER
// * Delete the control and unload dataset if they exists
//********************************************************************
//JOBLIB DD DSN=DB2A.SDSNLOAD,DISP=SHR
//DELETE EXEC PGM=IEFBR14
//DELREC00 DD DSN=V71A.SIEBEL.SYSRECO0,
//    UNIT=SYSDA,SPACE=(1024,(10,10)),DISP=(MOD,DELETE),
//    VOL=SER=DB0005
//DELREC01 DD DSN=V71A.SIEBEL.SYSRECO1,
//    UNIT=SYSDA,SPACE=(1024,(10,10)),DISP=(MOD,DELETE),
//    VOL=SER=DB0005
//DELPUNCH DD DSN=V71A.SIEBEL.SYSPUNCH,
//    UNIT=SYSDA,SPACE=(800,(15,15)),DISP=(MOD,DELETE),
//    VOL=SER=DB0005
//UNLOAD EXEC PGM=IKJEFT01,DYNAMNBR=20,COND=(4,LT)
//SYSTSPRT DD SYSOUT=*  
//SYSTSIN DD *
DSN SYSTEM(V71A)
RUN PROGRAM(DSNTIAUL) PLAN(DSNTIR71) -
    LIB('DB2A.TESTLIB')
//SYSPRINT DD SYSOUT=*  
//SYSUDUMP DD SYSOUT=*  
//SYSRECO0 DD DSN=V71A.SIEBEL.SYSRECO0,
//    UNIT=SYSDA,SPACE=(1024,(10,10)),DISP=(,CATLG),
//    VOL=SER=DB0005
//SYSRECO1 DD DSN=V71A.SIEBEL.SYSRECO1,
//    UNIT=SYSDA,SPACE=(1024,(10,10)),DISP=(,CATLG),
//    VOL=SER=DB0005
//SYSPUNCH DD DSN=V71A.SIEBEL.SYSPUNCH,
//    UNIT=SYSDA,SPACE=(800,(15,15)),DISP=(,CATLG),
//    VOL=SER=DB0005
//SYSSIN DD *
SIEBEL.TABLE  1530002
SIEBEL.TABLE2 1540002
/*
* EDIT THE OUTPUT FROM THE PROGRAM
* */
EXEC PGM=IKJEFT01,DYNAMNBR=20
SYSTSPRT DD SYSOUT=* 01550000
SYSTSIN DD * 01560000
EDIT 'V71A.SIEBEL.SYSPUNCH' DATA NONUM 01570000
CHANGE * 30 /SIEBEL.TABLE/USRT001.NEWTABLES/ 01580000
CHANGE * 30 /SIEBEL.TABLE2/USRT001.NEWTABLES2/ 01590000
TOP 01600000
LIST * 999 01610000
END SAVE 01620000
/*
* RUN LOAD UTILITY TO LOAD TABLES
* */
LOAD EXEC DSNUPROC,PARM='V71A,DSNTEX' 01630000
DSNTRACE DD SYSOUT=* 01640000
SORTLIB DD DSN=SYS1.SORTLIB,DISP=SHR 01650000
SORTWK01 DD UNIT=SYSDA,SPACE=(4000,(20,20),,,ROUND) 01660000
SORTWK02 DD UNIT=SYSDA,SPACE=(4000,(20,20),,,ROUND) 01670000
SORTWK03 DD UNIT=SYSDA,SPACE=(4000,(20,20),,,ROUND) 01680000
SORTWK04 DD UNIT=SYSDA,SPACE=(4000,(20,20),,,ROUND) 01690000
SYSREC00 DD DSN=V71A.SIEBEL.SYSREC00, 01700000
// DISP=(OLD,KEEP) 01710000
SYSREC01 DD DSN=V71A.SIEBEL.SYSREC01, 01720000
// DISP=(OLD,KEEP) 01730000
SYSUT1 DD UNIT=SYSDA,SPACE=(4000,(20,20),,,ROUND) 01740000
SYSIN DD DSN=V71A.SIEBEL.SYSPUNCH, 01750000
// DISP=(OLD,KEEP) 01760000
/*
REXX EXEC to generate DSNTIAUL unload jobs

/* rexx */
/* build DSNTIAUL UNLOAD jobs */
/* This job assumes a SMS dataclas and storclas */
/* called BIGDATA has been defined for use in */
/* the unload of the data. Compression and */
/* stripping should be considered for performance */
/* and economy of DASD space. */
/* */
/* input file is 5 columns */
/* dbname tsname owner tablename card */
/* */
/* select dbname,tsname,creator,name,card */
/* from sysibm.systables */
/* where type = 't' */
/* and card > 0 */
/* and creator='xxx' */
/* order by 1,2,3,4 */
/* */

trace OFF
signal on halt
/* you must update these values */
say "enter input dataset name"
parse upper pull inputname
if inputname="" then inputname="SQLOUT"
"ALLOC F(in) DA("inputname") SHR REUSE"

"EXECIO * DISKR in (STEM list. FINIS"
n = list.0
do idx = 1 to n
parse value list.idx with dbnm tsnm ownr tbnm card
  dbnm.idx = strip(dbnm)
  tsnm.idx = strip(tsnm)
  ownr.idx = strip(ownr)
  tbnm.idx = strip(tbnm)
  card.idx = strip(card)
end
/* create unload jobs */
drop cmd. /* initialize stem variable */
j = 1
k = 1
steps = 32
do i = 1 to n
  ii = right(i,4,'0')
  pp = (card.i%20) + 100
Appendix E. DB2 DSNTIAUL and LOAD jobs

```
smsclas = "DATACLAS=BIGDATA,STORCLAS=BIGDATA,"
units = ",3"
if pp > 72000 then units="4" /* >400 CYL */
if pp >108000 then units="5" /* >400 CYL */
if k = 1 then
do
  cmd.j ="//UNLD"ii" JOB ,                       ;j=j+1;
  cmd.j ="// MSGLEVEL=(1,1),MSGCLASS=H,REGION=OM,   ;j=j+1;
  cmd.j ="// TYPRUN=HOLD,CLASS=2,                  ;j=j+1;
  cmd.j ="// NOTIFY=&SYSUID,USER=&SYSUID,TIME=1440  ;j=j+1;
  cmd.j ="#*-------------------------------------------------";j=j+1;
  cmd.j ="//JOBLIB DD DISP=SHR,                     ;j=j+1;
  cmd.j ="//          DSN=DSN610.SDSNLOAD               ;j=j+1;
  cmd.j ="#*                                              ;j=j+1;
  end
  cmd.j="//D"ii" EXEC PGM=IDCAMS                     ;j=j+1;
  cmd.j="//SYSPRINT DD SYSOUT=*                      ;j=j+1;
  cmd.j="//SYSIN DD *                                 ;j=j+1;
  cmd.j=" DELETE (SIEBELF2.UNL"ii".SYSREC) NONVSAM  ;j=j+1;
  cmd.j=" DELETE (SIEBELF2.UNL"ii".SYSPUN) NONVSAM  ;j=j+1;
  cmd.j=" SET MAXCC=0                                  ;j=j+1;
  cmd.j="#*-------------------------------------------------";j=j+1;
  cmd.j="#*                                                 ;j=j+1;
  cmd.j=="/U"ii" EXEC PGM=IKJEFT01                    ;j=j+1;
  cmd.j="/SYSSTSPRT DD SYSOUT=*                       ;j=j+1;
  cmd.j="/SYSSTSN DD *                                ;j=j+1;
  cmd.j=" DSN SYSTEM(xxxx)                            ;j=j+1;
  cmd.j="/ RUN PROGRAM(DSNTIAUL) PLAN(DSNTIAUL) -     ;j=j+1;
  cmd.j=" LIB('xxxx.xxx.xx.RUNLIB.LOAD')              ;j=j+1;
  cmd.j="/SYSSTPRINT DD SYSOUT=*                     ;j=j+1;
  cmd.j="/SYSSTUDMP DD SYSOUT=*                      ;j=j+1;
  cmd.j="/SYSREC00 DD DSN=SIEBELF2.UNL"ii".SYSREC,   ;j=j+1;
  cmd.j="/ DISP=(,CATLG),UNIT=(3390"units")          ;j=j+1;
  cmd.j="/ "smsclas"                                 ;j=j+1;
  cmd.j="/ SPACE=(4096,"pp","pp"),RLSE)              ;j=j+1;
  cmd.j="/SYSSTPUNCH DD DSN=SIEBELF2.UNL"ii".SYSPUN,  ;j=j+1;
  cmd.j="/ DISP=(,CATLG),UNIT=3390,                  ;j=j+1;
  cmd.j="/ SPACE=(TRK,(1,1),RLSE)                    ;j=j+1;
  cmd.j="/SYSSTN DD *                                 ;j=j+1;
  cmd.j=" "ownr.i"."tbnm.i"                         ;j=j+1;
  cmd.j="/#*-------------------------------------------------";j=j+1;
  k = k + 1
  if k > steps then
do
    k = 1
    cmd.j="/E-O-J--------E-O-J--------E-O-J--------E-O-J---";j=j+1;
  end
end /* do i = 1 to n */
```
outputname = UNLOAD
x = SYSDSN(outputname)
if x <>ok then
  do
    say "allocating dataset"
    "ALLOC DA("outputname") DSORG(PS) UNIT(SYSDA) LRECL(80),
    BLKSIZE(4000) RECFM(F,B) SPACE(1,1) CYLINDERS"
  end
  "ALLOC F(out) DA("outputname") SHR REUSE"
  "EXECIO * DISKW out (STEM "cmd." FINIS"
  "ISPEXC Edit DATASET("outputname")"
  "FREE F(OUT)"
  /* stop reformat */

halt:
  "FREE F(IN)"
  "DELSTACK"
EXIT
REXX EXEC to generate DB2 LOAD jobs

/* rexx                                           */
/* build DB2 LOAD JOBS                            */
/* input file is 5 columns                        */
/*     dbname tsname owner tablename card        */
/*                                                */
/*    select dbname,tsname,owner,name,card        */
/*    from sysibm.systables                       */
/*      where   type = 't'                          */
/*      and   card > 0                            */
/*      and   owner = 'xxx'                       */
/*    order by 1,2,3,4                            */
/*                                                */
/*  edit syspunch via clist editlist and          */
/*  macro macrload in clist.fb                    */

trace OFF
signal on halt
/*  you must update these values                  */
say "enter input dataset name"
parse upper pull inputname
if inputname="" then inputname="SQLOUT"
"ALLOC F(in) DA("inputname") SHR REUSE"
"EXECIO * DISKR in (STEM list. FINIS"

n = list.0
do idx = 1 to n
    parse value list.idx with dbnm tsnm ownr tbnm card
    dbnm.idx   = strip(dbnm)
    tsnm.idx   = strip(tsnm)
    ownr.idx   = strip(ownr)
    tbnm.idx   = strip(tbnm)
    card.idx   = strip(card)
end
/* create unload jobs                            */
drop cmd. /* initialize stem variable            */
j = 1
k = 1
err=-1
steps = 32
do i = 1 to n
    ii = right(i,4,'0')
    if card.i >= 0 then pp=(card.i%100)+100
    else pp = 3600
    units = ""
    if pp > 18000 then units="",2" /* >100 CYL */
    if pp > 36000 then units="",3" /* >200 CYL */
if pp > 72000 then units="4"
/* >400 CYL */
if oldts ¬= tsnm.i then k=1
if (k = 1) then
do
err = err+1
err = right(err,3,'0')
cmd.j = "//tsnm.i" JOB ,
cmd.j = "MSGLEVEL=(1,1),MSGCLASS=H,REGION=OM,",
cmd.j = "TYPRUN=HOLD,CLASS=Z,
cmd.j = "NOTIFY=INIT,USER=&SYSUID,TIME=1440",
cmd.j = "/JOBPARM S=*",
cmd.j = "-------------------------------",
cmd.j = "//JObLIB DD DISP=SHR,DSN=DSNT1.DT11.SDSNLOAD",
cmd.j = "*-----------------------------------------------",
cmd.j = "//JOBLIB DD DISP=SHR,DSN=DSNT1.DT11.SDSNLOAD",
cmd.j = "*-----------------------------------------------",
cmd.j = "*                                             ",
cmd.j = "//ALLOC EXEC PGM=IEFBR14",
cmd.j = "DD1 DD DISP=(MOD,CATLG),",
cmd.j = "UNIT=SYSALLDA,SPACE=(CYL,(1,1)),",
cmd.j = "RECFM=FB,LRECL=80,BLKSIZE=8000",
cmd.j = "DSN=USERID.ERRerr",
cmd.j = "*                                             ",
cmd.j = "*-------------------------------------------------",
cmd.j = "*                                                ",
cmd.j = "L"ii" EXEC PGM=DSNUTLB,PARM='DT11,L"ii"
",
cmd.j = "SYSPRINT DD SYSOUT=*
",
cmd.j = "SYSUDUMP DD SYSOUT=*
",
cmd.j = "UTPRINT DD SYSOUT=*
",
cmd.j = "DSNTRACE DD SYSOUT=*
",
cmd.j = "SYSDISC DD SYSOUT=*
",
cmd.j = "SYSERR DD DISP=(,DELETE,DELETE),
",
cmd.j = "STORCLAS=DB2ARCH,
",
cmd.j = "SPACE=(CYL,(50,50)),UNIT=SYSALLDA,
",
cmd.j = "DSN=SIEBELF2.UNL"ii".SYSERR",
cmd.j = "STORCLAS=DB2ARCH,
",
cmd.j = "SPACE=(CYL,(50,50)),UNIT=SYSALLDA,
",
cmd.j = "DSN=SIEBELF2.UNL"ii".SYSMAP",
cmd.j = "STORCLAS=DB2ARCH,
",
cmd.j = "SPACE=(CYL,(50,50)),UNIT=SYSALLDA,
",
cmd.j = "DSN=SIEBELF2.UNL"ii".SORTOUT",
cmd.j = "STORCLAS=DB2ARCH,
",
cmd.j = "SPACE=(4096,"pp","pp"),UNIT=(SYSALLDAunits"
",
cmd.j = "DSN=SIEBELF2.UNL"ii".SORTOUT",
cmd.j = "STORCLAS=DB2ARCH,
",
cmd.j = "SPACE=(4096,"pp","pp"),UNIT=(SYSALLDAunits"
",
cmd.j = "DSN=SIEBELF2.UNL"ii".SYSUT1",
cmd.j = "STORCLAS=DB2ARCH,
",
cmd.j = "SPACE=(4096,"pp","pp"),UNIT=(SYSALLDAunits"
",
cmd.j = "DSN=SIEBELF2.UNL"ii".SORTWK01",
cmd.j = "STORCLAS=DB2ARCH,
",
cmd.j = "SPACE=(4096,"pp","pp"),UNIT=(SYSALLDAunits"
",
cmd.j = "DSN=SIEBELF2.UNL"ii".SORTWK02",
cmd.j = "STORCLAS=DB2ARCH,
",
cmd.j = "SPACE=(4096,"pp","pp"),UNIT=(SYSALLDAunits"
",
cmd.j = "DSN=SIEBELF2.UNL"ii".SORTWK03"
APPENDIX E. DB2 DSNTIAUL AND LOAD JOBS

```plaintext
//SORTWK04 DD UNIT=SYSALLDA,SPACE=(CYL,550) 
;j=j+1;
//SORTWK05 DD UNIT=SYSALLDA,SPACE=(CYL,550) 
;j=j+1;
//SORTWK06 DD UNIT=SYSALLDA,SPACE=(CYL,550) 
;j=j+1;
//SORTWK07 DD UNIT=SYSALLDA,SPACE=(CYL,550) 
;j=j+1;
//SORTWK08 DD UNIT=SYSALLDA,SPACE=(CYL,550) 
;j=j+1;
//SYSREC00 DD DISP=SHR,DSN=SIEBELF2.UNL"ii".SYSREC 
;j=j+1;
//SYSIN DD DISP=SHR,DSN=SIEBELF2.UNL"ii".SYSPUN 
;j=j+1;
//          DD *                                    
;j=j+1;
REPAIR SET TABLESPACE "dbnm.i"."tsnm.i" NOCOPYPEND 
;j=j+1;
//SYSIN     DD DISP=SHR,DSN=SIEBELF2.UNL"ii".SYSPUN 
;j=j+1;
//          DD *                                    
;j=j+1;
//SYSREC00 DD DISP=SHR,DSN=SIEBELF2.UNL"ii".SYSREC 
;j=j+1;
//SYSIN     DD DISP=SHR,DSN=SIEBELF2.UNL"ii".SYSPUN 
;j=j+1;
CMD.J="REPAIR SET TABLESPACE "dbnm.i"."tsnm.i" NOCOPYPEND"
;j=j+1;
IF "ii" IF (L"ii".RC>4 | L"ii".ABEND) THEN 
;j=j+1;
EXEC PGM=ICEGENER 
;j=j+1;
SYSIN DD DUMMY 
;j=j+1;
SYSPRINT DD SYSOUT=H 
;j=j+1;
---ERROR HISTORY DATASET---
;j=j+1;
SYSUT2 DD DISP=MOD,DSN=SIEBELF2.ERR"err" 
;j=j+1;
SYSUT1 DD * 
;j=j+1;
"L"ii" "dbnm.i"."tsnm.i" "ownr.i"."tbnm.i" 
;j=j+1;
ENDIF 
;j=j+1;
/*-E-O-J--------E-O-J--------E-O-J--------E-O-J---*/
j=j+1;
k = k + 1
oldts = tsnm.i
if k > steps then
  do
    k = 1
  cmd.j="*-*E-O-J--------E-O-J--------E-O-J--------E-O-J---"j=j+1;
end
/* do i = 1 to n */
outputname = RELOAD
x = SYSDSN(outputname)
if x ¬=ok then
  do
    say "allocating dataset"
    "ALLOC DA("outputname") DSORG(PS) UNIT(SYSALLDA) LRECL(80),
    BLKSIZE(4000) RECFM(F,B) SPACE(1,2) CYLINDERS"
  end
"ALLOC F(out) DA("outputname") SHR REUSE"
"EXECIO * DISKW out (STEM "cmd." FINIS"
"ISPEXEC EDIT DATASET("outputname")"
"FREE F(OUT)"
/* stop reformat */
halt:
"FREE F(IN)"
"DELSSTACK"
EXIT
```

Appendix E. DB2 DSNTIAUL and LOAD jobs
Validating object names

Table names > 18 characters

You can use the following SQL statement to search for table names that exceed 18 characters:

```
SELECT T.NAME "TABLE NAME"
,      LENGTH(T.NAME) "TBLNAMELN"
FROM S_TABLE T
WHERE T.REPOSITORY_ID = (SELECT ROW_ID FROM S_REPOSITORY
                          WHERE NAME = 'SIEBEL REPOSITORY'
                          )
   AND   LENGTH(T.NAME) > 18
ORDER BY T.NAME
```

The third column points to the converted name in DB2 for OS/390.

Index names > 18 characters

You can use the following SQL statement to search for index names that exceed 18 characters:

```
SELECT T.NAME "TABLE NAME"
,      I.NAME "INDEX NAME"
,      LENGTH(I.NAME) "IDXNAMELN"
FROM S_INDEX I
```

© Copyright IBM Corp. 2001
WHERE T.ROW_ID = I.TBL_ID
AND T.REPOSITORY_ID = (SELECT ROW_ID FROM S_REPOSITORY
WHERE NAME = 'SIEBEL REPOSITORY')
AND LENGTH(I.NAME) > 18
ORDER BY T.NAME,
        I.NAME

Column names > 18 characters

You can use the following SQL statement to search for column names that exceed 18 characters:

```
select T.NAME "Table Name",
       C.NAME "Column Name",
       length(C.NAME) "ColNameLn"
from S_TABLE T
    , S_COLUMN C
where T.ROW_ID = C.TBL_ID
and T.REPOSITORY_ID = (select ROW_ID from S_REPOSITORY
                       where NAME = 'Siebel Repository')
and length(C.NAME) > 18
order by T.NAME,
        C.NAME
```

Index key > 255 bytes

To find the index keys exceeding 255 bytes you may use the following SQL statement:

```
SELECT T.NAME "TABLE NAME",
       I.NAME "INDEX NAME",
       sum(case C.DATA_TYPE
            when 'C' then C.LENGTH
            when 'V' then C.LENGTH
            when 'D' then 7
            when 'S' then 7
            when 'N' then C.LENGTH/2
            when 'X' then 0
            end
       ) "390 LENGTH"
FROM S_TABLE T
    , S_COLUMN C
    , S_INDEX I
```
Appendix F. Validating object names

WHERE T.ROW_ID = C.TBL_ID
AND T.ROW_ID = I.TBL_ID
AND I.ROW_ID = IC.INDEX_ID
AND C.ROW_ID = IC.COL_ID
AND T.REPOSITORY_ID = (SELECT ROW_ID FROM S_REPOSITORY
WHERE NAME = 'SIEBEL REPOSITORY')

GROUP BY T.NAME
, I.NAME
HAVING SUM(CASE C.DATA_TYPE
    WHEN 'C' THEN C.LENGTH
    WHEN 'V' THEN C.LENGTH
    WHEN 'D' THEN 8
    WHEN 'S' THEN 8
    WHEN 'N' THEN C.LENGTH/2
    WHEN 'X' THEN 0
    END
) > 255

The .dat file

Example of .dat file for one table.

Siebel Data File V3
FILE_TYPE=000000000003
FILE_VERSION=000000000003
TotTbs=000000000001
TotRows=000000000784
TbdynA=000000000000

TABLE_NAME="S_DOC_AGREE"

NumRows=000000000784
numCols="110"
COL_NAME="ROW_ID"
COL_NAME="CREATED"
COL_NAME="CREATED_BY"
COL_NAME="LAST_UPD"
COL_NAME="LAST_UPD_BY"
COL_NAME="DCKING_NUM"
COL_NAME="MODIFICATION_NUM"
COL_NAME="CONFLICT_ID"
COL_NAME="NAME"
COL_NAME="REV_NUM"
COL_NAME="ACCTNT_SRC_ID"
COL_NAME="AGREE_ADDR_ID"
COL_NAME="AGREE_CD"
COL_NAME="AGREE_NUM"
<table>
<thead>
<tr>
<th>COL_NAME</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;AGREE_TYPE_CD&quot;</td>
<td></td>
</tr>
<tr>
<td>&quot;ALL_CON_FLG&quot;</td>
<td></td>
</tr>
<tr>
<td>&quot;ALL_PROD_FLG&quot;</td>
<td></td>
</tr>
<tr>
<td>&quot;AMENDMENT&quot;</td>
<td></td>
</tr>
<tr>
<td>&quot;APPR_BY_EMP_ID&quot;</td>
<td></td>
</tr>
<tr>
<td>&quot;APPR_DT&quot;</td>
<td></td>
</tr>
<tr>
<td>&quot;BILL_TO_ADDR_ID&quot;</td>
<td></td>
</tr>
<tr>
<td>&quot;BILL_TO_CON_ID&quot;</td>
<td></td>
</tr>
<tr>
<td>&quot;BK_AMT&quot;</td>
<td></td>
</tr>
<tr>
<td>&quot;BK_AMT_CURCY_CD&quot;</td>
<td></td>
</tr>
<tr>
<td>&quot;CNTR_RESP_TIME_HRS&quot;</td>
<td></td>
</tr>
<tr>
<td>&quot;CNTR_SRV_CAL_CD&quot;</td>
<td></td>
</tr>
<tr>
<td>&quot;CNTR_SRV_TYPE_CD&quot;</td>
<td></td>
</tr>
<tr>
<td>&quot;CNTR_SRV_TZ_CD&quot;</td>
<td></td>
</tr>
<tr>
<td>&quot;CON_PER_ID&quot;</td>
<td></td>
</tr>
<tr>
<td>&quot;CURRENT_QUOTA&quot;</td>
<td></td>
</tr>
<tr>
<td>&quot;CUST_DOC_STAT_CD&quot;</td>
<td></td>
</tr>
<tr>
<td>&quot;CUST_RFP_NUM&quot;</td>
<td></td>
</tr>
<tr>
<td>&quot;DESC_TEXT&quot;</td>
<td></td>
</tr>
<tr>
<td>&quot;DUE_DT&quot;</td>
<td></td>
</tr>
<tr>
<td>&quot;EFF_DATE&quot;</td>
<td></td>
</tr>
<tr>
<td>&quot;EFF_END_DT&quot;</td>
<td></td>
</tr>
<tr>
<td>&quot;EFF_START_DT&quot;</td>
<td></td>
</tr>
<tr>
<td>&quot;FILE_CD&quot;</td>
<td></td>
</tr>
<tr>
<td>&quot;FILE_NAME&quot;</td>
<td></td>
</tr>
<tr>
<td>&quot;INIT_QUOTA&quot;</td>
<td></td>
</tr>
<tr>
<td>&quot;INTEGRATION_ID&quot;</td>
<td></td>
</tr>
<tr>
<td>&quot;LEGAL_ADDR_ID&quot;</td>
<td></td>
</tr>
<tr>
<td>&quot;LEGAL_CON_ID&quot;</td>
<td></td>
</tr>
<tr>
<td>&quot;OPTY_ID&quot;</td>
<td></td>
</tr>
<tr>
<td>&quot;PAR_AGREE_ID&quot;</td>
<td></td>
</tr>
<tr>
<td>&quot;PAYMENT_TERM_ID&quot;</td>
<td></td>
</tr>
<tr>
<td>&quot;PO_AMT&quot;</td>
<td></td>
</tr>
<tr>
<td>&quot;PO_AMT_CURCY_CD&quot;</td>
<td></td>
</tr>
<tr>
<td>&quot;PO_AMT_DT&quot;</td>
<td></td>
</tr>
<tr>
<td>&quot;PRI_LST_ID&quot;</td>
<td></td>
</tr>
<tr>
<td>&quot;PROC_PRIORITY_CD&quot;</td>
<td></td>
</tr>
<tr>
<td>&quot;PROJ_ID&quot;</td>
<td></td>
</tr>
<tr>
<td>&quot;PR_REP_PER_ID&quot;</td>
<td></td>
</tr>
<tr>
<td>&quot;RENEWABLE_FLG&quot;</td>
<td></td>
</tr>
<tr>
<td>&quot;REV_DT&quot;</td>
<td></td>
</tr>
<tr>
<td>&quot;SALES_REP_POSTN_ID&quot;</td>
<td></td>
</tr>
<tr>
<td>&quot;SERVICE_FLG&quot;</td>
<td></td>
</tr>
<tr>
<td>&quot;SHIP_TO_ADDR_ID&quot;</td>
<td></td>
</tr>
<tr>
<td>&quot;SHIP_TO_CON_ID&quot;</td>
<td></td>
</tr>
<tr>
<td>&quot;SIGNATURE_DT&quot;</td>
<td></td>
</tr>
<tr>
<td>&quot;SIGNATURE_NAME&quot;</td>
<td></td>
</tr>
</tbody>
</table>
COL_NAME="STAT_CD"
COL_NAME="TARGET_OU_ID"
COL_NAME="TEMPLATE_FLG"
COL_NAME="TURN_AMT"
COL_NAME="TURN_AMT_CURCY_CD"
COL_NAME="TURN_AMT_DT"
COL_NAME="TURN_BY_DT"
COL_NAME="VALID_FLG"
COL_NAME="X_PAYFREQUENCY"
COL_NAME="X_PAYMECHANISM"
COL_NAME="X_PAYMETHOD"
COL_NAME="X_TAPEID"
COL_NAME="X_AGREE_FLG"
COL_NAME="X_CONTACT_FLG"
COL_NAME="X_SALE_FLG"
COL_NAME="ACTIVE_FLG"
COL_NAME="BL_CURCY_CD"
COL_NAME="BU_ID"
COL_NAME="DISCNT_PERCENT"
COL_NAME="EST_AMT_CURCY_CD"
COL_NAME="EST_AMT_EXCH_DT"
COL_NAME="EST_COST_AMT"
COL_NAME="EST_REVN_AMT"
COL_NAME="ORDER_ID"
COL_NAME="QUOTE_ID"
COL_NAME="RATE_LST_ID"
COL_NAME="SVC_PROVIDER_ID"
COL_NAME="X_AGENT_NAME"
COL_NAME="X_AGENT_SIGN_FLG"
COL_NAME="X_BATCH_NUMBER"
COL_NAME="X_CANCELLATION_REA"
COL_NAME="X_CONTRACT_STATUS_"
COL_NAME="X_COOL_OFF_DATE"
COL_NAME="X_CSV_FILENAME"
COL_NAME="X_DD_SIGN_DATE"
COL_NAME="X_DD_SIGN_FLG"
COL_NAME="X_DELETE_FLAG"
COL_NAME="X_ELEC_PREPAY_METE"
COL_NAME="X_GAS_PREPAY_METER"
COL_NAME="X_PHONE_AUDIT_FLG"
COL_NAME="X_RECORD_TYPE"
COL_NAME="X_SALES_ADVISOR_PI"
COL_NAME="X_SALES_CHANNEL"
COL_NAME="X_SALES_OFFICE_PIN"
COL_NAME="X_SIGN_DATE"
COL_NAME="X_SIGN_FLG"
COL_NAME="X_TIF_IMAGE_NUM"
{ROW_ID="1+1NG+1"}

Appendix F. Validating object names  173
**Dataexp/FTP/DB2 Load programs and jobs**

**REXX procedure**

```rexx
/*REXX */
ARG DB2 TBCREATOR
IF DB2 = '' THEN DO
   SAY 'ENTER DB2 SYSTEM NAME'
   PULL DB2
END
TSSUB = 0
TSNAMES = ''
PQYYS. = ''
IF TBCREATOR = '' THEN TBCREATOR = 'SIEBEL'
TDT = 'D\'SUBSTR(DATE(S),3)
CALL RXSUBCOM 'ADD', 'SQL', 'DSNREXX'
RXCD = RC
IF RXCD > 4 THEN DO
   SAY 'DB2 CONNECTION FAILED'
   SAY RXCD 'FROM RXSUBCOM'
END
ADDRESS SQL 'CONNECT' DB2
RXCD = RC
IF RXCD > 4 THEN DO
```

© Copyright IBM Corp. 2001
SAY 'DB2 CONNECTION FAILED'
SAY RXCD 'FROM DSNALI'
END
"EXECIO 1 DISKR COUNTS("
RCD = RC
DO WHILE RCD = 0
    CALL PROCESS_COUNTS
"EXECIO 1 DISKR COUNTS("
RCD = RC
END
"EXECIO 1 DISKR COUNTS(FINIS"
DO SUB = 1 TO WORDS(TSNAMES)
    PRIQTY = PQTYS.SUB
    IF PRIQTY < 1 THEN PRIQTY = 1
    SECQTY = PRIQTY % 10
    IF SECQTY < 1 THEN SECQTY = 1
    QUEUE 'ALTER TABLESPACE 'WORD(TSNAMES,SUB)
    QUEUE 'PRIQTY 'PRIQTY 'SECQTY 'SECQTY';'
    "EXECIO "QUEUED()" DISKW ALTERS("
END
"EXECIO "QUEUED()" DISKW ALTERS(FINIS"
ADDRESS SQL 'DISCONNECT' DB2
EXIT
PROCESS_COUNTS:
    PULL COUNTS
    TBNAME = WORD(COUNTS,1)
    ROWS = WORD(COUNTS,2)
    SEL = "SELECT A.TSNAME,A.DBNAME,A.RECLENGTH,B.PGSIZE "
    SEL = SEL "FROM SYSEBSYSTABLES A,SYSEBSYSTABLESPACE B"
    SEL = SEL " WHERE A.NAME = 'TBNAME'"
    SEL = SEL " AND A.CREATOR = 'TBCREATOR'"
    SEL = SEL " AND A.DBNAME = B.DBNAME"
    SEL = SEL " AND A.TSNAME = B.NAME"
    CSR = 'C1'
    SNUM = 'S1'
    CALL OPEN_CURSOR
ADDRESS SQL
"EXECSQL FETCH C1 INTO :TSNAME,:DBNAME,:RECLENGTH,:PGSIZE"
IF SQLCODE/=0 THEN DO
    SAY 'SQLCODE 'SQLCODE 'FOR 'TBCREATOR'.TBNAME
    SAY SEL
    EXIT 16
END
COUNT = 0
JNUM = 0
MNUM = 0
DO WHILE SQLCODE = 0
    ADDRESS TSO
    SAY TBNAME
BYTES_LEFT = (1024 * PGSIZE) - 22
RPAGE = BYTES_LEFT $ RECLENGTH
IF RPAGE > 255 THEN RPAGE = 255
PFREE = RPAGE $ 10
RPAGE = RPAGE - PFREE
NUMPAGES = ROWS $ RPAGE
NUMSP_PAGES = NUMPAGES $ 4800
NUMPAGES = NUMPAGES + NUMSP_PAGES
PRIQTY = NUMPAGES * PGSIZE
TSN = STRIP(DBNAME) .'STRIP(TSNAME)
FND = 'N'
FPOS = WORDPOS(TSN,TSNAMES)
IF FPOS = 0 THEN DO
  TSSUB = TSSUB + 1
  PQTYS.TSSUB = PRIQTY
  TSNAMES = TSNAMES TSN
END ELSE ,
PQTYS.FPOS = PQTYS.FPOS + PRIQTY
ADDRESS SQL
"EXEC SQL FETCH C1 INTO :TSNAME,:DBNAME,:RECLENGTH,:PGSIZE"
IF SQLCODE/=0 & SQLCODE ^= 100 THEN DO
  SAY 'SQLCODE 'SQLCODE
  EXIT 16
END
END
ADDRESS SQL
"EXEC SQL CLOSE "CSR
RETURN
OPEN_CURSOR:
ADDRESS SQL
"EXEC SQL DECLARE "CSR" CURSOR FOR "SNUM"
IF SQLCODE/=0 THEN DO
  SAY 'SQLCODE 'SQLCODE
  EXIT 16
END
ADDRESS SQL
"EXEC SQL PREPARE "SNUM" FROM :SEL"
IF SQLCODE/=0 THEN DO
  SAY 'SQLCODE 'SQLCODE
  EXIT 16
END
ADDRESS SQL
"EXEC SQL OPEN "CSR
IF SQLCODE/=0 THEN DO
  SAY 'SQLCODE 'SQLCODE
  EXIT 16
END
RETURN
JCL to run the REXX procedure

//FLETCHPE JOB (541), 'PAUL FLETCHER', CLASS=A, MSGCLASS=H, NOTIFY=FLETCHP
//*
//* THIS JOB WILL RUN THE SIEBEL SIZING REXX
//*
//* SIEBSIZE EXEC PGM=IKJEFT01, DYNAMNBR=180, REGION=OM
//*
//* STEPLIB DD DSN=SYS2.DB2.V610.SDSNLOAD, DISP=SHR
//* SYSEXEC DD DSN=FLETCHP.MASTER.REXX, DISP=SHR
//* SYSIN DD *
//*
//* SYSTSPRT DD SYSOUT=* 
//* SYSOUT DD SYSOUT=* 
//* COUNTS DD DSN=FLETCHP.SIEBL.COUNTSX, DISP=SHR 
//* ALTERS DD DSN=FLETCHP.DB2.INPUT(SIEBALTS), DISP=SHR 
//* SYSDUMP DD SYSOUT=* 
//* SYSIN DD * 
*SIEBSIZE ISC1 SIEBL
/*
PLFSIEBL COBOL program

IDENTIFICATION DIVISION.

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

PROGRAM-ID. PLFSIEBL.

AUTHOR. PAUL FLETCHER.

ENVIRONMENT DIVISION.

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

CONFIGURATION SECTION.

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

SOURCE-COMPUTER.

IBM-370.

INPUT-OUTPUT SECTION.

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

FILE-CONTROL.

SELECT INPUT-FILE
  ASSIGN UT-S-SIEBELIN
  FILE STATUS IS WG-SIEBEL-FILE-STATUS.

SELECT OUTPUT-FILE
  ASSIGN UT-S-DDSEQ01W
  FILE STATUS IS WG-SIEBEL-FILE-STATUS.

SELECT LOADCDS-FILE
  ASSIGN UT-S-LOADCDS
  FILE STATUS IS WG-SIEBEL-FILE-STATUS.

I-O-CONTROL.

  APPLY WRITE-ONLY OUTPUT-FILE.

DATA DIVISION.

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

FILE SECTION.

**************
FD    INPUT-FILE
   LABEL RECORDS ARE STANDARD
   RECORDING MODE IS F
   BLOCK CONTAINS 0
   DATA RECORDS IS SIEBEL-REC.

 01 SIEBEL-REC    PIC X(80).

FD    OUTPUT-FILE
   LABEL RECORDS ARE STANDARD
   RECORDING MODE IS V
   RECORD IS VARYING IN SIZE
       DEPENDING ON W400-OUT-LEN
   BLOCK CONTAINS 0
   DATA RECORDS IS SIEBEL-OUT.

 01 SIEBEL-OUT.
   03 SIEBEL-BYTE    PIC X OCCURS 32714
      DEPENDING ON W400-OUT-LEN.

FD    LOADCDS-FILE
   LABEL RECORDS ARE STANDARD
   RECORDING MODE IS F
   BLOCK CONTAINS 0
   DATA RECORDS IS SIEBEL-REC.

 01 LOADCDS-REC    PIC X(80).

WORKING-STORAGE SECTION.
***************************************************************************
 *
***************************************************************************

*   GENERAL WORKING STORAGE
 *
 01 FILLER             PIC XXX VALUE 'WG-'.
 01 FILLER.
   03 W100-SHORT-MSG   PIC X(24).
   03 W100-LONG-MSG    PIC X(400).
   03 W100-SELECT      PIC X(8) VALUE 'SELECT '.
   03 W100-ISPLINK     PIC X(8) VALUE 'ISPLINK '.
   03 W100-CMD-LEN     PIC S9(9) COMP VALUE +21.
   03 W100-ADDRESS     PIC S9(9) COMP.
   03 W100-CMD.
05 FILLER PIC X(12) VALUE 'CMD(DB2SPRX '.
05 W100-HEX-ADDRESS PIC X(8).
05 FILLER PIC X VALUE '
.
03 W200-SQLCODE PIC -9(4).
01 W400-OUTPUT-AREA.
  03 W400-OUT-LEN PIC 9(8) COMP VALUE 132.
  03 W400-OUTPUT-REC.
  05 W400-OUTPUT-BYTE PIC X
     OCCURS 32714 DEPENDING ON W400-OUT-LEN.

01 WD-DB2-CONNECT.
  03 WD-FUNCTION PIC X(12).
  03 WD-SSNAME PIC X(4) VALUE 'ISC1'.
  03 WD-TERMBCB PIC X(4).
  03 WD-STARTCB PIC X(4).
  03 WD-RIBPTR PIC X(4).
  03 WD-RET-CODE PIC S9(8) COMP.
  03 WD-REASCODE PIC S9(8) COMP.
  03 WD-PLAN PIC X(8) VALUE 'PLFSIEBL'.
  03 WD-TERMOP PIC X(4) VALUE 'SYNC'.

01 WG-GENERAL-STORAGE.
  03 WG-TSO-PARM1 PIC S9(9) COMP VALUE +0.
  03 WG-TSO-RETCODE PIC S9(8) COMP VALUE +0.
  03 WG-TSO-REASCODE PIC S9(8) COMP VALUE +0.
  03 WG-TSO-ABENDCODE PIC S9(8) COMP VALUE +0.
  03 WG-TSO-CPPL-ADDR PIC S9(8) COMP VALUE +0.
  03 WG-MAX-RECLENGTH PIC S9(4) COMP VALUE +0.
  03 WG-INDVAR PIC S9(4) COMP.
  03 WG-SIEBEL-RECORD PIC X(80).
  03 WG-SIEBEL-BUFF PIC X(80).
  03 WG-PDS-NAME PIC X(50).
  03 WG-PDS-NAME-LEN PIC S9(3) COMP-3 VALUE +0.
  03 WG-DIV-ZERO PIC S9(3) COMP-3 VALUE +0.
  03 WG-COLFOUND PIC X VALUE 'N'.
  03 WG-COLUMN-TYPE PIC X(8).
  88 VARIABLE-COLUMN VALUES
    'VARCHAR',
    'LONGVAR',
    'VARG',
    'LONGVARG'.
  03 WG-REC-COUNT PIC S9(17) COMP-3 VALUE +0.
  03 WG-RECS-WRITTEN PIC S9(17) COMP-3 VALUE +0.
  03 WG-SUB2 PIC S9(4) COMP VALUE +0.
  03 WG-SUB3 PIC S9(8) COMP VALUE +0.
  03 WG-SUB PIC S9(4) COMP VALUE +0.
  03 WG-COLSUB PIC S9(4) COMP VALUE +0.
  03 WG-COLNUM PIC S9(3) COMP-3 VALUE +0.
03 WG-COLNAM PIC X(18).
03 WG-LOAD-TYPE PIC X(8).
03 WG-BUFF-POS PIC S9(4) COMP VALUE +0.
03 WG-NEGATIVE PIC X.
03 WG-REC-POS PIC S9(4) COMP VALUE +0.
03 WG-NUM-COLS PIC 9(3).
03 WG-NUM-TABLES PIC S9(4) COMP VALUE +0.
03 WG-NUM-TABR REDEFINES WG-NUM-TABLES PIC XX.
03 WG-NUM-ROWS PIC S9(8) COMP VALUE +0.
03 WG-NUM-ROWR REDEFINES WG-NUM-ROWS PIC XXXX.
03 WG-HEX PIC S9(4) COMP VALUE +0.
03 WG-HEX-R REDEFINES WG-HEX PIC XX.
03 WG-WLEN PIC S9(4) COMP VALUE +0.
03 WG-TOT-LEN PIC S9(4) COMP VALUE +0.
03 WG-TBCREATOR-LEN PIC S9(3) COMP-3 VALUE +0.
03 WG-TBCREATOR PIC X(8).
03 WG-LEN-LEFT PIC S9(4) COMP VALUE +0.
03 WG-DEC-LEFT PIC S9(3) COMP-3 VALUE +0.
03 WG-LNAME PIC X(8).
03 WG-LNAME-LEN PIC S9(3) COMP-3 VALUE 0.
03 WG-RECLENGTH PIC S9(4) COMP.
03 WG-RECSTORE PIC S9(4) COMP.
03 WG-DECSTORE PIC S9(4) COMP.
03 WG-DEC-STORE-R REDEFINES WG-DEC-STORE PIC XX.
03 WG-DEC-POS PIC S9(3) COMP-3.
03 WG-DEC-STORE PIC S9(4) COMP.
03 WG-DEC-STORE-R REDEFINES WG-DEC-STORE.
03 WG-DEC-MOVE PIC S9(4) COMP.
03 WG-DEC-MOVE-R REDEFINES WG-DEC-MOVE.
05 WG-DECSTORE1 PIC X.
05 WG-DECSTORE2 PIC X.
03 WG-DBNAME PIC X(8).
Appendix G. Dataexp/FTP/DB2 Load programs and jobs

03 WG-OBID PIC S9(4) COMP.
03 WG-OBID-R REDEFINES WG-OBID PIC XX.
03 WG-ROWNUM PIC S9(4) COMP VALUE +0.
03 WG-ROWNUM-R REDEFINES WG-ROWNUM.
   05 FILLER PIC X.
   05 WG-ROWNUM2 PIC X.
03 WG-PSIZE PIC S9(4) COMP.
03 WG-ROWS-PER-PAGE PIC S9(3) COMP-3.
03 WG-DB2-DSNAME.
   49 WG-DB2-DSN-LEN PIC S9(4) COMP.
   49 WG-DB2-DSN PIC X(52).
03 WG-TOT-COLS PIC S9(3) COMP-3.
03 WG-NUM-TSNAMES PIC S9(3) COMP-3 VALUE +0.
03 WG-TSNAMES OCCURS 750 PIC X(8).
03 WG-COL-XLAT OCCURS 750 PIC S9(4) COMP.
03 WG-COL-DETAILS OCCURS 750.
   05 WG-COLNAME PIC X(18).
   05 WG-COLNAME-LEN PIC S9(4) COMP.
   05 WG-COLTYPE PIC X(8).
   05 WG-COLLENGTH PIC S9(4) COMP.
   05 WG-SCALE PIC S9(4) COMP.
   05 WG-NULLS PIC X.
   05 WG-COL-IN-USE PIC X.
   05 WG-START-DISPL PIC S9(4) COMP.

01 W700-IKJEFTSR PIC X(8) VALUE 'IKJEFTSR'.
01 W700-IKJEFTSR-P1 PIC S9(8) COMP VALUE +65541.
01 W700-IKJEFTSR-CMD-LEN PIC S9(8) COMP VALUE +0.
01 W700-IKJEFTSR-RETCODE PIC S9(8) COMP VALUE +0.
01 W700-IKJEFTSR-REASCODE PIC S9(8) COMP VALUE +0.
01 W700-IKJEFTSR-ABNDCODE PIC S9(8) COMP VALUE +0.
01 W700-TSO-COMMAND PIC X(180).

01 WX-PARM1.
   03 WX-LEN PIC S9(4) COMP VALUE 4.
   03 WX-CODE PIC S9(8) COMP.

01 WX-PARM2.
   03 WX-HEX-DISPLAY PIC X(8).

EXEC SQL
   INCLUDE SYSTABLE
END-EXEC.

EXEC SQL
   INCLUDE SYSCOLS
END-EXEC.
EXEC SQL
  INCLUDE SQLCA
END-EXEC.

EXEC SQL
  DECLARE SYSCOLS CURSOR FOR
  SELECT A.NAME, A.COLTYPE,
      A.LENGTH, A.SCALE, A.NULLS, B.DBNAME, B.OBID,
      B.TSNAME, B.RECLENGTH
  FROM SYIBM.SYSCOLUMNS A,
      SYIBM.SYSTABLES B
  WHERE A.TBCREATOR = :WSYS-TBCREATOR
      AND A.TBNAME = :WSYS-TBNAME
      AND B.NAME = A.TBNAME
      AND B.CREATOR = A.TBCREATOR
  ORDER BY COLNO
END-EXEC.

LINKAGE SECTION.
*****************
01 PARM-INPUT PIC X(80).

PROCEDURE DIVISION USING PARM-INPUT.
*****************
**

A000-MAINLINE SECTION.
  ACCEPT PARM-INPUT.
  CALL 'IKJTSOEV' USING WG-TSO-PARM1
      WG-TSO-RETCODE
      WG-TSO-REASCODE
      WG-TSO-ABENDCODE
      WG-TSO-CPPL-ADDR.
  IF WG-TSO-RETCODE NOT = 0
      AND WG-TSO-RETCODE NOT = 24
      DISPLAY 'RETCODE FROM IKJTSOEV = ' WG-TSO-RETCODE
      GOBACK
  END-IF.
  DISPLAY 'PARM = ' PARM-INPUT.
  PERFORM A100-GET-NEXT-WORD.
  PERFORM UNTIL WG-WLEN = 0
      IF PARM-INPUT (WG-START:WG-WLEN) = 'DB2ID'
      PERFORM A100-GET-NEXT-WORD
MOVE PARM-INPUT (WG-START:WG-WLEN) TO WD-SSNAME
ELSE
IF PARM-INPUT (WG-START:WG-WLEN) = 'LOAD'
    PERFORM A100-GET-NEXT-WORD
    MOVE PARM-INPUT (WG-START:WG-WLEN) TO WG-LOAD-TYPE
ELSE
IF PARM-INPUT (WG-START:WG-WLEN) = 'CREATER'
    PERFORM A100-GET-NEXT-WORD
    MOVE PARM-INPUT (WG-START:WG-WLEN) TO WSYS-TBCREATOR
    WG-TBCREATOR
    MOVE WG-WLEN TO WG-TBCREATOR-LEN
END-IF
END-IF
END-IF
PERFORM A100-GET-NEXT-WORD
END-PERFORM.
PERFORM X700-DB2-CONNECT.
OPEN INPUT INPUT-FILE.
IF WG-SIEBEL-FILE-STATUS > '09'
    DISPLAY 'SIEBEL FILE OPEN ERROR ' WG-SIEBEL-FILE-STATUS
    GOBACK
END-IF.
PERFORM R000-READ.
MOVE WG-SIEBEL-BUFF TO WG-SIEBEL-RECORD
MOVE WG-SIEBEL-RECORD (41:2) TO WG-NUM-TABR
DISPLAY 'NUMBER OF TABLES ' WG-NUM-TABLES
MOVE 53 TO WG-BUFF-POS
PERFORM WG-NUM-TABLES TIMES
    MOVE 0 TO WG-RECS-WRITTEN
    PERFORM E000-GET-LENGTH
    PERFORM E100-GET-DATA
    MOVE 1 TO WG-SUB
    MOVE WG-LEN TO W-NAME-LEN
    ADD WG-LEN TO WG-SUB
    GIVING WG-END
    PERFORM X100-CONVERT-TO-EBCDIC
    MOVE WG-LEN TO WSYS-TBNAME-LEN
    MOVE WG-SIEBEL-RECORD (1:WG-LEN)
    TO WSYS-TBNAME-TEXT
    DISPLAY WSYS-TBNAME-TEXT
    PERFORM B100-GET-TABLE-DETAILS
    MOVE 6 TO WG-LEN
    PERFORM E100-GET-DATA
    MOVE WG-SIEBEL-RECORD (3:4)
    TO WG-NUM-ROWR
    DISPLAY 'WG-NUM-ROWS = ' WG-NUM-ROWS
    PERFORM E000-GET-LENGTH
    PERFORM E100-GET-DATA
    MOVE 1 TO WG-SUB
SUBTRACT 1 FROM WG-END
PERFORM X100-CONVERT-TO-EBCDIC
MOVE WG-SIEBEL-RECORD (1:WG-LEN) TO WG-NUM-COLS
DISPLAY 'WG-NUM-COLS = ' WG-NUM-COLS
MOVE 0 TO WG-COLNUM
PERFORM C000-PROCESS-COLNAMES WG-NUM-COLS TIMES
PERFORM X200-ALLOCATE-FILE
PERFORM D000-PROCESS-DATA WG-NUM-ROWS TIMES
CLOSE OUTPUT-FILE
DISPLAY 'RECORDS WRITTEN = ' WG-RECS-WRITTEN
MOVE 8 TO WG-LEN
PERFORM E100-GET-DATA
END-PERFORM.
CLOSE INPUT-FILE.
PERFORM X800-DB2-DISCONNECT
GOBACK.

A100-GET-NEXT-WORD.
********************************
ADD 1 TO WG-SUB3.
PERFORM VARYING WG-SUB3 FROM WG-SUB3 BY 1
UNTIL PARM-INPUT (WG-SUB3:1) NOT = SPACE
OR WG-SUB3 > 80
END-PERFORM.
IF WG-SUB3 > 80
MOVE 0 TO WG-WLEN
ELSE
MOVE WG-SUB3 TO WG-START
ADD 1 TO WG-SUB3
PERFORM VARYING WG-SUB3 FROM WG-SUB3 BY 1
UNTIL PARM-INPUT (WG-SUB3:1) = SPACE
OR WG-SUB3 > 80
END-PERFORM
SUBTRACT WG-START FROM WG-SUB3 GIVING WG-WLEN
END-IF.
DISPLAY PARM-INPUT (WG-START:WG-WLEN).

B100-GET-TABLE-DETAILS SECTION.
EXEC SQL
OPEN SYSCOLS
END-EXEC.
IF SQLCODE NOT = 0
MOVE SQLCODE TO W200-SQLCODE
MOVE 'OPEN SYSCOLS FAILURE' TO W100-SHORT-MSG
MOVE 'SQLCODE ' TO W100-LONG-MSG
MOVE W200-SQLCODE TO W100-LONG-MSG (13:8)
DISPLAY W100-SHORT-MSG
DISPLAY W100-LONG-MSG
GOBACK
END-IF.

EXEC SQL
FETCH SYSCOLS INTO
:WSYS-NAME,:WSYS-COLTYPE,
:WSYS-LENGTH,:WSYS-SCALE,:WSYS-NULLS,
:W-DBNAME,:W-OBID,
:W-TSNAME,:W-RECLENGTH
END-EXEC.

IF SQLCODE NOT = 0
  MOVE SQLCODE TO W200-SQLCODE
  MOVE 'FETCH SYSCOLS FAILURE' TO W100-SHORT-MSG
  MOVE 'SQLCODE ' TO W100-LONG-MSG
  MOVE W200-SQLCODE TO W100-LONG-MSG (13:8)
  DISPLAY W100-SHORT-MSG
  DISPLAY W100-LONG-MSG
  GOBACK
END-IF.

MOVE 0 TO WG-SUB.
MOVE 7 TO WG-START-DISPL (1)
PERFORM UNTIL SQLCODE NOT = 0
  ADD 1 TO WG-SUB
  MOVE WSYS-NAME-TEXT (1:WSYS-NAME-LEN) TO WG-COLNAME (WG-SUB)
  MOVE WSYS-NAME-LEN TO WG-COLNAME-LEN (WG-SUB)
  MOVE WSYS-COLTYPE TO WG-COLTYPE (WG-SUB)
  IF WSYS-COLTYPE = 'DECIMAL'
    DIVIDE WSYS-LENGTH BY 2 GIVING WSYS-LENGTH
    ADD 1 TO WSYS-LENGTH
    GIVING WSYS-LENGTH
  END-IF
  MOVE WSYS-LENGTH TO WG-COLLENGTH (WG-SUB)
  MOVE WSYS-SCALE TO WG-SCALE (WG-SUB)
  MOVE WSYS-NULLS TO WG-NULLS (WG-SUB)
  ADD 1 TO WG-SUB GIVING WG-SUB2
  ADD WSYS-LENGTH TO WG-START-DISPL (WG-SUB2)
  GIVING WG-START-DISPL (WG-SUB2)
  IF WG-NULLS (WG-SUB) = 'Y'
    ADD 1 TO WG-START-DISPL (WG-SUB2)
  END-IF
  MOVE WSYS-COLTYPE TO WG-COLUMN-TYPE
  IF VARIABLE-COLUMN
    ADD 2 TO WG-START-DISPL (WG-SUB2)
  END-IF
  IF WG-SUB = 1
    MOVE W-DBNAME TO W-DBNAME
  END-IF
MOVE W-OBID TO WG-OBID
MOVE X'02' TO W400-OUTPUT-REC (1:1)
MOVE W-TSNAME TO WG-TSNAME
PERFORM VARYING WG-TSNAME-LEN FROM 8 BY -1
UNTIL WG-TSNAME (WG-TSNAME-LEN:1) NOT = SPACE
END-PERFORM
MOVE WG-OBID-R TO W400-OUTPUT-REC (4:2)
END-IF
EXEC SQL
  FETCH SYSCOLS INTO
    :WSYS-NAME,:WSYS-COLTYPE,
    :WSYS-LENGTH,:WSYS-SCALE,:WSYS-NULLS,
    :W-DBNAME,:W-OBID,
    :W-TSNAME,:W-RECLENGTH
END-EXEC
IF SQLCODE NOT = 0
  AND SQLCODE NOT = 100
  MOVE SQLCODE TO W200-SQLCODE
  MOVE 'FETCH SYSCOLS FAILURE' TO W100-SHORT-MSG
  MOVE 'SQLCODE ' TO W100-LONG-MSG
  MOVE W200-SQLCODE TO W100-LONG-MSG (13:8)
  DISPLAY W100-SHORT-MSG
  DISPLAY W100-LONG-MSG
  GOBACK
END-IF.
EXEC SQL
  CLOSE SYSCOLS
END-EXEC.
IF SQLCODE NOT = 0
  MOVE SQLCODE TO W200-SQLCODE
  MOVE 'CLOSE SYSCOLS FAILURE' TO W100-SHORT-MSG
  MOVE 'SQLCODE ' TO W100-LONG-MSG
  MOVE W200-SQLCODE TO W100-LONG-MSG (13:8)
  DISPLAY W100-SHORT-MSG
  DISPLAY W100-LONG-MSG
  GOBACK
END-IF.
SUBTRACT 1 FROM WG-START-DISPL (WG-SUB2)
GIVING WG-RECLENGTH
MOVE WG-RECLENGTH TO WG-RECLENGTH
  W400-OUT-LEN
MOVE WG-RECLENGTH-R TO W400-OUTPUT-REC (2:2)
MOVE WG-SUB TO WG-TOT-COLS.
EXEC SQL
  SELECT PGSIZE INTO :WG-PGSIZE
  FROM SYSIBM.SYSTABLESPACE
  WHERE DBNAME = :W-DBNAME
  AND NAME = :W-TSNAME
END-EXEC.
IF SQLCODE NOT = 0
  MOVE SQLCODE TO W200-SQLCODE
  MOVE 'SELECT SYSTSPACE FAILURE' TO W100-SHORT-MSG
  MOVE 'SQLCODE ' TO W100-LONG-MSG
  MOVE W200-SQLCODE TO W100-LONG-MSG (13:8)
  DISPLAY W100-SHORT-MSG
  DISPLAY W100-LONG-MSG
  GOBACK
END-IF.
MULTIPLY WG-PGSIZE BY 1024 GIVING WG-PGSIZE.
DIVIDE WG-PGSIZE BY WG-RECLENGTH
GIVING WG-ROWS-PER-PAGE.

C000-PROCESS-COLNAMES SECTION.
  PERFORM E000-GET-LENGTH
  PERFORM E100-GET-DATA
  MOVE 1 TO WG-SUB.
  MOVE WG-LEN TO WG-END.
  PERFORM X100-CONVERT-TO-EBCDIC.
  MOVE 'N' TO WG-COLFOUND.
  ADD 1 TO WG-COLNUM.
  PERFORM VARYING WG-COLSUB FROM 1 BY 1
  UNTIL WG-COLFOUND = 'Y' OR WG-COLSUB > WG-TOT-COLS
  IF WG-LEN > 18
    DISPLAY '**** ERROR ' W100-SIEBEL-RECORD (1:WG-LEN)
    ' GREATER THAN 18 CHARACTERS'
    MOVE 18 TO WG-LEN
  END-IF
  IF WG-LEN = WG-COLNAME-LEN (WG-COLSUB) AND
     W100-SIEBEL-RECORD (1:WG-LEN) =
     WG-COLNAME (WG-COLSUB)
    MOVE 'Y' TO WG-COLFOUND
    WG-COL-IN-USE (WG-COLSUB)
    DISPLAY W100-SIEBEL-RECORD (1:WG-LEN) ' FOUND'
    MOVE WG-COLSUB TO WG-COL-XLAT (WG-COLNUM)
  END-IF
END-PERFORM
IF WG-COLFOUND = 'N'
  DISPLAY 'COLUMN NAME ' W100-SIEBEL-RECORD (1:WG-LEN)
  ' NOT FOUND IN DB2 DEFINITION OF TABLE' WSYS-TBNAME
  GOBACK
END-IF.

D000-PROCESS-DATA SECTION.
  SUBTRACT WG-BUFF-POS FROM 81 GIVING WG-END
  IF WG-END = 0
PERFORM R000-READ
    MOVE 3 TO WG-BUFF-POS
ELSE
    IF WG-END = 1
        PERFORM R000-READ
        MOVE 2 TO WG-BUFF-POS
    ELSE
        IF WG-END = 2
            PERFORM R000-READ
            MOVE 1 TO WG-BUFF-POS
        ELSE
            ADD 2 TO WG-BUFF-POS
        END-IF
    END-IF
END-IF.

PERFORM D100-PROCESS-COLUMNS VARYING WG-SUB FROM 1 BY 1
UNTIL WG-SUB > WG-NUM-COLS.
PERFORM VARYING WG-SUB2 FROM 1 BY 1
UNTIL WG-SUB2 > WG-TOT-COLS
    IF WG-COL-IN-USE (WG-SUB2) NOT = 'Y'
        MOVE -1 TO WG-LEN
    END-IF
END-PERFORM
ADD 1 TO WG-ROWNUM.
IF WG-ROWNUM > WG-ROWS-PER-PAGE
    MOVE 1 TO WG-ROWNUM
END-IF
MOVE WG-ROWNUM2 TO W400-OUTPUT-REC (6:1)
WRITE SIEBEL-OUT FROM W400-OUTPUT-REC
IF WG-SIEBEL-FILE-STATUS > '00'
    DISPLAY 'SIEBEL WRITE ERROR ' WG-SIEBEL-FILE-STATUS
    GOBACK
END-IF.
ADD 1 TO WG-RECS-WRITTEN.

D100-PROCESS-COLUMNS SECTION.
    MOVE WG-COL-XLAT (WG-SUB) TO WG-SUB2
    PERFORM E000-GET-LENGTH.
    PERFORM D200-MOVE-DATA.

D200-MOVE-DATA SECTION.
    MOVE WG-START-DISPL (WG-SUB2) TO WG-SUB3.
    MOVE WG-COLTYPE (WG-SUB2) TO WG-COLUMN-TYPE.
    IF VARIABLE-COLUMN
        IF WG-LEN = -1
            MOVE 1 TO WG-LEN
            MOVE WG-LEN-R TO W400-OUTPUT-REC (WG-SUB3:2)
ADD 2 TO WG-SUB3
MOVE HIGH-VALUES TO W400-OUTPUT-REC (WG-SUB3:1)
ADD 1 TO WG-SUB3
MOVE WG-COLLENGTH (WG-SUB2) TO WG-LEN
MOVE LOW-VALUES TO W400-OUTPUT-REC (WG-SUB3:WG-LEN)
ELSE
  IF WG-NULLS (WG-SUB2) = 'Y'
    ADD 1 TO WG-LEN
END-IF
MOVE WG-LEN-R TO W400-OUTPUT-REC (WG-SUB3:2)
ADD 2 TO WG-SUB3
IF WG-NULLS (WG-SUB2) = 'Y'
  MOVE LOW-VALUES TO W400-OUTPUT-REC (WG-SUB3:1)
  ADD 1 TO WG-SUB3
  SUBTRACT 1 FROM WG-LEN
END-IF
PERFORM D800-MOVE-CHAR
END-IF
ELSE
  IF WG-COLUMN-TYPE = 'CHAR'
    IF WG-LEN = -1
      MOVE HIGH-VALUES TO W400-OUTPUT-REC (WG-SUB3:1)
      ADD 1 TO WG-SUB3
      MOVE WG-COLLENGTH (WG-SUB2) TO WG-LEN
      MOVE LOW-VALUES TO W400-OUTPUT-REC (WG-SUB3:WG-LEN)
    ELSE
      IF WG-NULLS (WG-SUB2) = 'Y'
        MOVE LOW-VALUES TO W400-OUTPUT-REC (WG-SUB3:1)
        ADD 1 TO WG-SUB3
      END-IF
      PERFORM D800-MOVE-CHAR
    END-IF
  ELSE
    IF WG-COLUMN-TYPE = 'DECIMAL'
      PERFORM D300-DECIMAL-COLUMN
    ELSE
      IF WG-COLUMN-TYPE = 'TIMESTAMP'
        PERFORM D400-TIMESTAMP-COLUMN
      END-IF
    END-IF
  END-IF
ELSE
  IF WG-COLUMN-TYPE = 'DATE'
    PERFORM D300-DATE-COLUMN
  ELSE
  END-IF
END-IF.

D300-DECIMAL-COLUMN SECTION.
  IF WG-LEN = -1
    MOVE HIGH-VALUES TO W400-OUTPUT-REC (WG-SUB3:1)
    ADD 1 TO WG-SUB3
    MOVE WG-COLLENGTH (WG-SUB2) TO WG-LEN
    DIVIDE WG-LEN BY 2
GIVING WG-LEN
ADD 1 TO WG-LEN
MOVE LOW-VALUES TO W400-OUTPUT-REC (WG-SUB3:WG-LEN)
ELSE
PERFORM E100-GET-DATA
IF WG-NULLS (WG-SUB2) = 'Y'
  MOVE LOW-VALUES TO W400-OUTPUT-REC (WG-SUB3:1)
  ADD 1 TO WG-SUB3
END-IF
MULTIPLY WG-COLLENGTH (WG-SUB2) BY 2
GIVING WG-DEC-LEN
PERFORM VARYING WG-DEC-POS FROM 1
  BY 1 UNTIL WG-SIEBEL-RECORD (WG-DEC-POS:1) = X'2E'
OR WG-DEC-POS > WG-LEN
END-PERFORM
IF WG-DEC-POS > WG-LEN
  MOVE WG-LEN TO WG-DEC-LEFT
  MOVE 0 TO WG-DEC-RIGHT
ELSE
  SUBTRACT WG-DEC-POS FROM WG-LEN
  GIVING WG-DEC-RIGHT
  SUBTRACT 1 FROM WG-DEC-POS
  GIVING WG-DEC-LEFT
END-IF
SUBTRACT WG-SCALE (WG-SUB2) FROM WG-DEC-LEN
GIVING WG-TEMP
SUBTRACT WG-DEC-LEFT FROM WG-TEMP
GIVING WG-START
ADD 1 TO WG-START
MOVE 1 TO WG-REC-POS
IF WG-SIEBEL-RECORD (1:1) = X'2D'
  MOVE 208 TO WG-DEC-MOVE
ELSE
  MOVE 240 TO WG-DEC-MOVE
END-IF
MOVE '2' TO WG-NEXT
PERFORM VARYING WG-DEC-POS FROM 2 BY 1
UNTIL WG-DEC-POS > WG-DEC-LEN
IF WG-SIEBEL-RECORD (WG-REC-POS:1) = X'2E'
  ADD 1 TO WG-START
  WG-REC-POS
ELSE
  IF WG-SIEBEL-RECORD (WG-REC-POS:1) = X'2D'
    ADD 1 TO WG-START
    WG-REC-POS
  ELSE
    IF WG-NEXT = '1'
      IF WG-START = WG-DEC-POS
        MOVE WG-SIEBEL-RECORD (WG-REC-POS:1)
TO WG-DECM2
SUBTRACT 48 FROM WG-DEC-MOVE
MULTIPLY WG-DEC-MOVE BY 16
GIVING WG-DEC-MOVE
ADD 1 TO WG-START
WG-REC-POS
END-IF
MOVE '2' TO WG-NEXT
ELSE
IF WG-START = WG-DEC-POS
MOVE WG-SIEBEL-RECORD (WG-REC-POS:1)
TO WG-DECSTORE2
ADD 1 TO WG-REC-POS
SUBTRACT 48 FROM WG-DEC-STORE
ADD WG-DEC-STORE TO WG-DEC-MOVE
ADD 1 TO WG-START
END-IF
MOVE WG-DECM2
TO W400-OUTPUT-REC (WG-SUB3:1)
ADD 1 TO WG-SUB3
MOVE '1' TO WG-NEXT
MOVE 0 TO WG-DEC-MOVE
WG-DEC-STORE
END-IF
END-IF
END-IF
END-PERFORM
END-IF.

D400-TIMESTAMP-COLUMN SECTION.
IF WG-LEN = -1
MOVE HIGH-VALUES TO W400-OUTPUT-REC (WG-SUB3:1)
ADD 1 TO WG-SUB3
MOVE WG-COLLENGTH (WG-SUB2) TO WG-LEN
MOVE LOW-VALUES TO W400-OUTPUT-REC (WG-SUB3:WG-LEN)
ELSE
MOVE 0 TO WG-DEC-MOVE
PERFORM E100-GET-DATA
IF WG-NULLS (WG-SUB2) = 'Y'
MOVE LOW-VALUES TO W400-OUTPUT-REC (WG-SUB3:1)
ADD 1 TO WG-SUB3
END-IF
MOVE 1 TO WG-NEXT
PERFORM VARYING WG-DEC-POS FROM 1 BY 1
UNTIL WG-DEC-POS > WG-LEN
IF WG-SIEBEL-RECORD (WG-DEC-POS:1) = X'2D'
MOVE NEXT TO WG-NEXT
ELSE
IF WG-SIEBEL-RECORD (WG-DEC-POS:1) = X'20'
    MOVE WG-NEXT TO WG-NEXT
ELSE
    IF WG-SIEBEL-RECORD (WG-DEC-POS:1) = X'2E'
        MOVE WG-NEXT TO WG-NEXT
    ELSE
        IF WG-SIEBEL-RECORD (WG-DEC-POS:1) = X'3A'
            MOVE WG-NEXT TO WG-NEXT
        ELSE
            IF WG-NEXT = '1'
                MOVE 0 TO WG-DEC-MOVE
                MOVE WG-SIEBEL-RECORD (WG-DEC-POS:1)
                    TO WG-DECM2
                SUBTRACT 48 FROM WG-DEC-MOVE
                MULTIPLY WG-DEC-MOVE BY 16
                GIVING WG-DEC-MOVE
                MOVE '2' TO WG-NEXT
            ELSE
                MOVE 0 TO WG-DEC-STORE
                MOVE WG-SIEBEL-RECORD (WG-DEC-POS:1)
                    TO WG-DECSTORE2
                SUBTRACT 48 FROM WG-DEC-MOVE
                MOVE '1' TO WG-NEXT
                ADD WG-DEC-STORE TO WG-DEC-MOVE
                MOVE WG-DECM2
                    TO W400-OUTPUT-REC (WG-SUB3:1)
                ADD 1 TO WG-SUB3
            END-IF
        END-IF
    END-IF
END-IF
END-IF
END-IF
END-PERFORM
IF WG-NEXT = '2'
    MOVE WG-DECM2
        TO W400-OUTPUT-REC (WG-SUB3:1)
    ADD 1 TO WG-SUB3
END-IF
IF WG-LEN = 19
    MOVE LOW-VALUES TO W400-OUTPUT-REC (WG-SUB3:3)
    ADD 3 TO WG-SUB3
END-IF
END-IF.

D800-MOVE-CHAR SECTION.
MOVE WG-LEN TO WG-TOT-LEN
SUBTRACT WG-BUFF-POS FROM 81
GIVING WG-LEN
IF WG-TOT-LEN NOT > WG-LEN
Appendix G. Dataexp/FTP/DB2 Load programs and jobs

MOVE WG-SIEBEL-BUFF (WG-BUFF-POS:WG-TOT-LEN) TO W400-OUTPUT-REC (WG-SUB3:WG-TOT-LEN)
ADD WG-TOT-LEN TO WG-BUFF-POS
ELSE
MOVE WG-SIEBEL-BUFF (WG-BUFF-POS:WG-LEN) TO W400-OUTPUT-REC (WG-SUB3:WG-LEN)
ADD WG-LEN TO WG-SUB3
MOVE WG-TOT-LEN TO WG-LEN-LEFT
PERFORM R000-READ
MOVE 1 TO WG-BUFF-POS
PERFORM UNTIL WG-LEN-LEFT < 1
SUBTRACT WG-BUFF-POS FROM 81 GIVING WG-LEN
IF WG-LEN-LEFT NOT > WG-LEN
MOVE WG-SIEBEL-BUFF (WG-BUFF-POS:WG-LEN-LEFT) TO W400-OUTPUT-REC (WG-SUB3:WG-LEN-LEFT)
ADD WG-LEN-LEFT TO WG-BUFF-POS
ELSE
MOVE WG-SIEBEL-BUFF (WG-BUFF-POS:WG-LEN) TO W400-OUTPUT-REC (WG-SUB3:WG-LEN)
ADD WG-LEN TO WG-SUB3
SUBTRACT WG-LEN FROM WG-LEN-LEFT
PERFORM R000-READ
END-IF
END-PERFORM
SUBTRACT WG-TOT-LEN FROM WG-COLLENGTH (WG-SUB2) GIVING WG-LEN
ADD 1 TO WG-LEN
MOVE LOW-VALUES TO W400-OUTPUT-REC (WG-SUB3:WG-LEN)
ADD 1 TO WG-SUB3
END-IF.

E000-GET-LENGTH SECTION.
SUBTRACT WG-BUFF-POS FROM 80 GIVING WG-END.
IF WG-END < 0
PERFORM R000-READ
MOVE WG-SIEBEL-BUFF (1:2) TO WG-LEN-R
MOVE 3 TO WG-BUFF-POS
ELSE
IF WG-END = 0
MOVE WG-SIEBEL-BUFF (80:1) TO WG-SIEBEL-RECORD
PERFORM R000-READ
MOVE WG-SIEBEL-BUFF (1:1) TO WG-SIEBEL-RECORD(2:1)
MOVE WG-SIEBEL-RECORD (1:2) TO WG-LEN-R
MOVE 2 TO WG-BUFF-POS
ELSE

Appendix G. Dataexp/FTP/DB2 Load programs and jobs 195
IF WG-END = 1
  MOVE WG-SIEBEL-BUFF (79:2) TO WG-SIEBEL-RECORD
  PERFORM R000-READ
  MOVE WG-SIEBEL-RECORD (1:2) TO WG-LEN-R
  MOVE 1 TO WG-BUFF-POS
ELSE
  MOVE WG-SIEBEL-BUFF (WG-BUFF-POS:2) TO WG-SIEBEL-RECORD
  MOVE WG-SIEBEL-RECORD (1:2) TO WG-LEN-R
  ADD 2 TO WG-BUFF-POS
END-IF
END-IF.

E100-GET-DATA SECTION.
SUBTRACT WG-BUFF-POS FROM 81 GIVING WG-END.
IF WG-LEN < WG-END
  MOVE WG-SIEBEL-BUFF (WG-BUFF-POS:WG-LEN) TO WG-SIEBEL-RECORD
  ADD WG-LEN TO WG-BUFF-POS GIVING WG-BUFF-POS
ELSE
  MOVE WG-SIEBEL-BUFF (WG-BUFF-POS:WG-END) TO WG-SIEBEL-RECORD
  PERFORM R000-READ
  ADD 1 TO WG-END GIVING WG-START
  SUBTRACT WG-END FROM WG-LEN GIVING WG-END
  MOVE WG-SIEBEL-BUFF (1:WG-END) TO WG-SIEBEL-RECORD (WG-START:WG-END)
  ADD 1 TO WG-END GIVING WG-BUFF-POS
END-IF.

R000-READ SECTION.
READ INPUT-FILE INTO WG-SIEBEL-BUFF
AT END MOVE 'Y' TO WG-EOF.
ADD 1 TO WG-REC-COUNT.

X100-CONVERT-TO-EBCDIC SECTION.
MOVE 0 TO WG-HEX.
PERFORM VARYING WG-SUB FROM WG-SUB BY 1
UNTIL WG-SUB > WG-END
  MOVE WG-SIEBEL-RECORD (WG-SUB:1) TO WG-HEX-R (2:1)
  IF WG-HEX > 47 AND WG-HEX < 58
    ADD 192 TO WG-HEX
  ELSE
    IF WG-HEX > 96 AND WG-HEX < 106
      ADD 96 TO WG-HEX
    ELSE
IF WG-HEX > 105 AND WG-HEX < 115
  ADD 103 TO WG-HEX
ELSE
  IF WG-HEX > 114 AND WG-HEX < 123
    ADD 111 TO WG-HEX
  ELSE
    IF WG-HEX > 64 AND WG-HEX < 74
      ADD 128 TO WG-HEX
    ELSE
      IF WG-HEX > 73 AND WG-HEX < 83
        ADD 135 TO WG-HEX
      ELSE
        IF WG-HEX > 82 AND WG-HEX < 91
          ADD 143 TO WG-HEX
        ELSE
          IF WG-HEX = 95
            ADD 14 TO WG-HEX
          END-IF
        END-IF
      END-IF
    END-IF
  END-IF
END-IF
END-IF
END-IF
END-IF
END-IF
END-IF
END-IF
END-PERFORM.

X200-ALLOCATE-FILE SECTION.
MOVE 'FREE F(DDSEQ01W)' TO W700-TSO-COMMAND
MOVE 16
  TO W700-IKJEFTSR-CMD-LEN
DISPLAY W700-TSO-COMMAND (1:W700-IKJEFTSR-CMD-LEN)
CALL W700-IKJEFTSR USING W700-IKJEFTSR-P1
  W700-TSO-COMMAND
  W700-IKJEFTSR-CMD-LEN
  W700-IKJEFTSR-RETCODE
  W700-IKJEFTSR-REASCODE
  W700-IKJEFTSR-ABNDCODE
MOVE 'FREE F(LOADCDS)' TO W700-TSO-COMMAND
MOVE 15
  TO W700-IKJEFTSR-CMD-LEN
DISPLAY W700-TSO-COMMAND (1:W700-IKJEFTSR-CMD-LEN)
CALL W700-IKJEFTSR USING W700-IKJEFTSR-P1
  W700-TSO-COMMAND
  W700-IKJEFTSR-CMD-LEN
  W700-IKJEFTSR-RETCODE
  W700-IKJEFTSR-REASCODE
  W700-IKJEFTSR-ABNDCODE
PERFORM VARYING WG-SUB FROM 1 BY 1
UNTIL WG-SUB > WG-NUM-TSNAMES
OR WG-TSNAMES (WG-SUB) = WG-TSNAME
END-PERFORM.
IF WG-SUB > WG-NUM-TSNAMES
MOVE WG-TSNAME TO WG-TSNAMES (WG-SUB)
PERFORM X300-ALLOC-NEWFILES
ADD 1 TO WG-NUM-TSNAMES
ELSE
MOVE 'ALOC F(DOSEQO1W) DA(' TO W700-TSO-COMMAND
MOVE WG-TSNAME (1:WG-TSNAME-LEN)
TO W700-TSO-COMMAND (22:WG-TSNAME-LEN)
MOVE 22
TO W700-IKJEFTSR-CMD-LEN
ADD WG-TSNAME-LEN
TO W700-IKJEFTSR-CMD-LEN
MOVE '.LOADFILE) MOD'
TO W700-TSO-COMMAND (W700-IKJEFTSR-CMD-LEN:14)
ADD 15 TO W700-IKJEFTSR-CMD-LEN
DISPLAY W700-TSO-COMMAND (1:W700-IKJEFTSR-CMD-LEN)
CALL W700-IKJEFTSR USING W700-IKJEFTSR-P1
W700-TSO-COMMAND
W700-IKJEFTSR-CMD-LEN
W700-IKJEFTSR-RETCODE
W700-IKJEFTSR-REASCODE
W700-IKJEFTSR-ABNDCODE
OPEN OUTPUT OUTPUT-FILE
DISPLAY 'OPEN EXECUTED'
MOVE 'ALOC F(LOADCDS) DA(' TO W700-TSO-COMMAND
MOVE WG-TSNAME (1:WG-TSNAME-LEN)
TO W700-TSO-COMMAND (21:WG-TSNAME-LEN)
MOVE 21
TO W700-IKJEFTSR-CMD-LEN
ADD WG-TSNAME-LEN
TO W700-IKJEFTSR-CMD-LEN
MOVE '.LOADCDS) MOD'
TO W700-TSO-COMMAND (W700-IKJEFTSR-CMD-LEN:14)
ADD 14 TO W700-IKJEFTSR-CMD-LEN
DISPLAY W700-TSO-COMMAND (1:W700-IKJEFTSR-CMD-LEN)
CALL W700-IKJEFTSR USING W700-IKJEFTSR-P1
W700-TSO-COMMAND
W700-IKJEFTSR-CMD-LEN
W700-IKJEFTSR-RETCODE
W700-IKJEFTSR-REASCODE
W700-IKJEFTSR-ABNDCODE
OPEN OUTPUT LOADCDS-FILE
MOVE 'INTO TABLE '
TO WG-LOAD-CARD
MOVE WG-TBCREATOR TO WG-LOAD-CARD (12:WG-TBCREATOR-LEN)
MOVE 12 TO WG-WLEN
ADD WG-TBCREATOR-LEN TO WG-WLEN
MOVE "." TO WG-LOAD-CARD (WG-WLEN:1)
ADD 1 TO WG-WLEN
MOVE WSYS-TBNAME-TEXT (1:WSYS-TBNAME-LEN)
TO WG-LOAD-CARD (WG-WLEN:WSYS-TBNAME-LEN)
ADD WSYS-TBNAME-LEN TO WG-WLEN
WRITE LOADCDS-REC FROM WG-LOAD-CARD
CLOSE LOADCDS-FILE
END-IF.

X300-ALLOC-NEWFILES SECTION.
MOVE 'ALLOC F(DSEQ01W) DA(' TO W700-TSO-COMMAND.
MOVE WG-TSNAME (1:WG-TSNAME-LEN)
TO W700-TSO-COMMAND (22:WG-TSNAME-LEN).
MOVE 22
TO W700-IKJEFTSR-CMD-LEN
ADD WG-TSNAME-LEN
TO W700-IKJEFTSR-CMD-LEN
MOVE '.LOADFILE) NEW'
TO W700-TSO-COMMAND (W700-IKJEFTSR-CMD-LEN:14).
ADD 15 TO W700-IKJEFTSR-CMD-LEN
MOVE ' LRECL(32718) BLKSIZE(32722) CYLINDERS SPACE(10,5)'.
TO W700-TSO-COMMAND (W700-IKJEFTSR-CMD-LEN:50)
ADD 50 TO W700-IKJEFTSR-CMD-LEN
MOVE ' RECFM(V B) BUFNO(20)'
TO W700-TSO-COMMAND (W700-IKJEFTSR-CMD-LEN:11)
ADD 21 TO W700-IKJEFTSR-CMD-LEN
DISPLAY W700-TSO-COMMAND (1:W700-IKJEFTSR-CMD-LEN)
CALL W700-IKJEFTSR USING W700-IKJEFTSR-P1
W700-TSO-COMMAND
W700-IKJEFTSR-CMD-LEN
W700-IKJEFTSR-RETCODE
W700-IKJEFTSR-REASCODE
W700-IKJEFTSR-ABNDCODE.

IF RETURN-CODE NOT = 0
DISPLAY 'RETURN-CODE ' RETURN-CODE
DISPLAY 'ERROR CALLING REXX CODE'
' RETCODE ' W700-IKJEFTSR-RETCODE
' ABEND CODE ' W700-IKJEFTSR-ABNDCODE
' REASON ' W700-IKJEFTSR-REASCODE
GOBACK
END-IF.
DISPLAY 'FILE ALLOCATED'
OPEN OUTPUT OUTPUT-FILE.
DISPLAY 'OPEN EXECUTED'
IF WG-SIEBEL-FILE-STATUS > '00'
DISPLAY 'SIEBEL OPEN ERROR ' WG-SIEBEL-FILE-STATUS
GOBACK
END-IF.
MOVE 'DELETE ' TO W700-TSO-COMMAND
MOVE WG-TSNAME (1:WG-TSNAME-LEN) TO W700-TSO-COMMAND (8:WG-TSNAME-LEN).
MOVE 8 TO W700-IKJEFTSR-CMD-LEN
ADD WG-TSNAME-LEN TO W700-IKJEFTSR-CMD-LEN
MOVE '.LOADCDS' TO W700-TSO-COMMAND (W700-IKJEFTSR-CMD-LEN:8).
ADD 8 TO W700-IKJEFTSR-CMD-LEN
DISPLAY W700-TSO-COMMAND (1:W700-IKJEFTSR-CMD-LEN)
CALL W700-IKJEFTSR USING W700-IKJEFTSR-P1
    W700-TSO-COMMAND
    W700-IKJEFTSR-CMD-LEN
    W700-IKJEFTSR-RETCODE
    W700-IKJEFTSR-REASCODE
    W700-IKJEFTSR-ABNDCODE.

MOVE 'ALLOC F(LOADCDS) DA(' TO W700-TSO-COMMAND.
MOVE WG-TSNAME (1:WG-TSNAME-LEN) TO W700-TSO-COMMAND (22:WG-TSNAME-LEN).
MOVE 22 TO W700-IKJEFTSR-CMD-LEN
ADD WG-TSNAME-LEN TO W700-IKJEFTSR-CMD-LEN
MOVE '.LOADCDS) NEW' TO W700-TSO-COMMAND (W700-IKJEFTSR-CMD-LEN:14).
ADD 14 TO W700-IKJEFTSR-CMD-LEN
MOVE ' LRECL(80) BLKSIZE(3200) TRACKS SPACE(1,1)' TO W700-TSO-COMMAND (W700-IKJEFTSR-CMD-LEN:50)
ADD 50 TO W700-IKJEFTSR-CMD-LEN
MOVE ' RECFM(F B)' TO W700-TSO-COMMAND (W700-IKJEFTSR-CMD-LEN:11)
ADD 11 TO W700-IKJEFTSR-CMD-LEN
DISPLAY W700-TSO-COMMAND (1:W700-IKJEFTSR-CMD-LEN)
CALL W700-IKJEFTSR USING W700-IKJEFTSR-P1
    W700-TSO-COMMAND
    W700-IKJEFTSR-CMD-LEN
    W700-IKJEFTSR-RETCODE
    W700-IKJEFTSR-REASCODE
    W700-IKJEFTSR-ABNDCODE.

IF RETURN-CODE NOT = 0
    DISPLAY 'RETURN-CODE ' RETURN-CODE
    DISPLAY 'ERROR CALLING REXX CODE'
    ' RETCODE ' W700-IKJEFTSR-RETCODE
ABEND CODE ' W700-IKJEFTSR-ABNDCODE
REASON ' W700-IKJEFTSR-REASCODE
GOBACK
END-IF.
DISPLAY 'FILE ALLOCATED'
OPEN OUTPUT LOADCDS-FILE.
DISPLAY 'OPEN EXECUTED'
IF WG-SIEBEL-FILE-STATUS = '00'
DISPLAY 'LOADCDS OPEN ERROR ' WG-SIEBEL-FILE-STATUS
GOBACK
END-IF.
IF WG-LOAD-TYPE = 'REPLACE'
MOVE 'LOAD RESUME NO REPLACE ' TO WG-LOAD-CARD
ELSE
MOVE 'LOAD RESUME ' TO WG-LOAD-CARD
END-IF
WRITE LOADCDS-REC FROM WG-LOAD-CARD
MOVE 'STATISTICS TABLE(ALL) INDEX(ALL) UPDATE ALL'
TO WG-LOAD-CARD
WRITE LOADCDS-REC FROM WG-LOAD-CARD
MOVE 'LOG NO FORMAT UNLOAD ASCII '
TO WG-LOAD-CARD
WRITE LOADCDS-REC FROM WG-LOAD-CARD
MOVE 'INTO TABLE '
TO WG-LOAD-CARD
MOVE WG-TBCREATOR TO WG-LOAD-CARD (12:WG-TBCREATOR-LEN)
MOVE 12 TO WG-WLEN
ADD WG-TBCREATOR-LEN TO WG-WLEN
MOVE '.' TO WG-LOAD-CARD (WG-WLEN:1)
ADD 1 TO WG-WLEN
MOVE WSYS-TBNAME-TEXT (1:WSYS-TBNAME-LEN)
TO WG-LOAD-CARD (WG-WLEN:WSYS-TBNAME-LEN)
ADD WSYS-TBNAME-LEN TO WG-WLEN
WRITE LOADCDS-REC FROM WG-LOAD-CARD
CLOSE LOADCDS-FILE
IF WG-SIEBEL-FILE-STATUS > '00'
DISPLAY 'LOADCDS WRITE ERROR ' WG-SIEBEL-FILE-STATUS
GOBACK
END-IF.

X700-DB2-CONNECT SECTION.
*************************
*****************************************************************************
*** CONNECT TO DB2 ***
*****************************************************************************
MOVE 'CONNECT' TO WD-FUNCTION.
MOVE SPACES TO WD-TERMCB
WD-STARTCB
WD-RIIBPTR.
CALL 'DSNALI' USING WD-FUNCTION
  WD-SSNAME
  WD-TERMCB
  WD-STARTCB
  WD-RIBPTR
  WD-RET-CODE
  WD-REASCODE.
IF WD-RET-CODE = 200
  NEXT SENTENCE
ELSE
  IF WD-RET-CODE NOT = 0
    * IF WD-RET-CODE = 200
    * PERFORM X800-DB2-DISCONNECT
    * END-IF
    MOVE WD-REASCODE TO WX-CODE
    CALL 'PLFDISPX' USING WX-PARM1
      WX-PARM2
    MOVE SQLCODE TO W200-SQLCODE
    MOVE 'CAF CONNECT FAILURE' TO W100-SHORT-MSG
    MOVE 'REASON CODE ' TO W100-LONG-MSG
    MOVE WX-HEX-DISPLAY TO W100-LONG-MSG (13:8)
    DISPLAY W100-SHORT-MSG
    DISPLAY W100-LONG-MSG
    GOBACK
    END-IF.
  END-IF.
MOVE 'OPEN' TO WD-FUNCTION.
CALL 'DSNALI' USING WD-FUNCTION
  WD-SSNAME
  WD-PLAN
  WD-RET-CODE
  WD-REASCODE.
IF WD-RET-CODE = 200
  NEXT SENTENCE
ELSE
  IF WD-RET-CODE NOT = 0
    MOVE WD-REASCODE TO WX-CODE
    CALL 'PLFDISPX' USING WX-PARM1
      WX-PARM2
    MOVE SQLCODE TO W200-SQLCODE
    MOVE 'CAF OPEN FAILURE' TO W100-SHORT-MSG
    MOVE 'REASON CODE ' TO W100-LONG-MSG
    MOVE WX-HEX-DISPLAY TO W100-LONG-MSG (13:8)
    DISPLAY W100-SHORT-MSG
    DISPLAY W100-LONG-MSG
    GOBACK
    END-IF
  END-IF.
END-IF.
X700-CONNECT-EXIT.
EXIT.

X800-DB2-DISCONNECT SECTION.
*******************************
*******************************************************************************
*** DISCONNECT
*******************************************************************************
*** DISCONNECT ***
*******************************************************************************

MOVE 'CLOSE' TO WD-FUNCTION.
MOVE 'SYNC' TO WD-TERMOP.
CALL 'DSNALI' USING WD-FUNCTION
    WD-TERMOP
    WD-RET-CODE
    WD-REASCODE.
IF WD-RET-CODE NOT = 0
    MOVE WD-REASCODE TO WX-CODE
    CALL 'PLFDISPX' USING WX-PARM1
        WX-PARM2
    MOVE SQLCODE TO W200-SQLCODE
    MOVE 'CAF CLOSE FAILURE' TO W100-SHORT-MSG
    MOVE 'REASON CODE ' TO W100-LONG-MSG
    MOVE WX-HEX-DISPLAY TO W100-LONG-MSG (13:8)
    DISPLAY W100-SHORT-MSG
    DISPLAY W100-LONG-MSG
END-IF.

MOVE 'DISCONNECT' TO WD-FUNCTION.
CALL 'DSNALI' USING WD-FUNCTION
    WD-RET-CODE
    WD-REASCODE.
IF WD-RET-CODE NOT = 0
    MOVE WD-REASCODE TO WX-CODE
    CALL 'PLFDISPX' USING WX-PARM1
        WX-PARM2
    MOVE SQLCODE TO W200-SQLCODE
    MOVE 'CAF DISCONNECT FAILURE' TO W100-SHORT-MSG
    MOVE 'REASON CODE ' TO W100-LONG-MSG
    MOVE WX-HEX-DISPLAY TO W100-LONG-MSG (13:8)
    DISPLAY W100-SHORT-MSG
    DISPLAY W100-LONG-MSG
END-IF.

X800-DISCONNECT-EXIT.
EXIT.
JCL to run the PLFSIEBL program

```
//FLETCHPA JOB (946),PAUL,MSGCLASS=H,NOTIFY=FLETCHP
//PLFSIEBL EXEC PGM=PLFSIEBL,REGION=0M
//*******************************************************************************
//* *
//*******************************************************************************
//STEPLIB  DD DISP=SHR,DSN=SYS2.DB2.V610.SDSNLOAD
00012000
//   DD DISP=SHR,DSN=FLETCHP.MASTER.LOAD
//   DD DISP=SHR,DSN=SYS1.LPALIB
//SIEBELIN DD DISP=SHR,DSN=FLETCHP.DADEXP.FILE
//SYSPRINT DD SYSOUT=*  
//SYSOUT DD SYSOUT=*  
//SYSUDUMP DD SYSOUT=*  
//SYSABEND DD SYSOUT=*  
//SYSTSPRT DD SYSOUT=*  
//SYSSIN DD *
DB2ID ISC1 LOAD REPLACE CREATOR SIEBEL  
/*
```

The SYSIN DD contains details essential to the running of the program. They must all be on one line, and are grouped in pairs—but the pairs can be in any order.

The first pair is DB2ID ISC1. It is used to allow the program to use Call Attach to connect to DB2 system ISC1.

If LOAD REPLACE is specified, then the program will generate LOAD REPLACE cards. If LOAD RESUME is specified, then LOAD RESUME cards will be generated.

The final pair is CREATOR SIEBEL. This gives the name of the table owner for the SIEBEL tables. It is used to obtain the column names for each table and to generate the Load cards.

Notice that no output files are specified in the JCL; instead, each file is dynamically created by the program.

- The load cards will be in a file called tablespacename.LOADCDS.
- The load files will be called tablespacename.LOADFILE.

Both of these files are prefixed with the userid of the job submitter.
Appendix G. Dataexp/FTP/DB2 Load programs and jobs

JCL to run the DB2 LOAD utility

```
//FLETCHP7 JOB (541), 'PAUL FLETCHER', MSGCLASS=H,
//          NOTIFY=FLETCHP
//*
//* LOAD THE DATA
//*
//*
//LOAD    EXEC DSNUPROC, SYSTEM=ISC1, UID=FLETCHP7,
//        COND=(8,LT)
//*
//STEPLIB DD DSN=SYS2.DB2.V610.ISC.SDSNLOAD, DISP=SHR
//SYSUT1 DD SPACE=(CYL,(100,10)), UNIT=SYSDA
//SORTOUT DD SPACE=(CYL,(100,10)), UNIT=SYSDA
//SYSREC DD DISP=SHR, DSN=FLETCHP.FFFFF001.LOADFILE
//SYSIN DD DISP=SHR, DSN=FLETCHP.FFFFF001.LOADCDS
//SYSABEND DD SYSOUT=*  
//SYSUDUMP DD SYSOUT=*  
//```
Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

IBM Redbooks

For information on ordering these publications see “How to get IBM Redbooks” on page 208.

- Siebel 2000 Database Implementation on OS/390 Using NT Siebel Servers, SG24-5953
- Implementing Siebel eBusiness Applications with DB2 UDB on AIX/NT, SG24-6211
- DB2 UDB Server for OS/390 and z/OS Version 7 - Presentation Guide, SG24-6121

Other resources

These publications are also relevant as further information sources:

- DB2 UDB for OS/390 and z/OS Utility Guide and Reference Version 7, SC26-9945

Referenced Web sites

These Web sites are also relevant as further information sources:

How to get IBM Redbooks

Search for additional Redbooks or redpieces, view, download, or order hardcopy from the Redbooks Web Site

ibm.com/redbooks

Also download additional materials (code samples or diskette/CD-ROM images) from this Redbooks site.

Redpieces are Redbooks in progress; not all Redbooks become redpieces and sometimes just a few chapters will be published this way. The intent is to get the information out much quicker than the formal publishing process allows.

IBM Redbooks collections

Redbooks are also available on CD-ROMs. Click the CD-ROMs button on the Redbooks Web Site for information about all the CD-ROMs offered, updates and formats.
Special notices

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Corporation, Dept. 600A, Mail Drop 1329, Somers, NY 10589 USA.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Any pointers in this publication to external Web sites are provided for convenience only and do not in any manner serve as an endorsement of these Web sites.
The following terms are trademarks of other companies:

C-bus is a trademark of Corollary, Inc. in the United States and/or other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and/or other countries.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States and/or other countries.

PC Direct is a trademark of Ziff Communications Company in the United States and/or other countries and is used by IBM Corporation under license.

ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States and/or other countries.

UNIX is a registered trademark in the United States and other countries licensed exclusively through The Open Group.

SET, SET Secure Electronic Transaction, and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC.

Other company, product, and service names may be trademarks or service marks of others.
Index

A
architecture 2

C
configurations 2
connected client 28
setup 17
connections 16
  DB2 Connect EE 16
  DDF 16
cross-loader 50

D
data 5
  repository 5
  seed 6, 26
  user 6
data warehousing tools 7
database
  connections 16
  migration 2, 3
  structure 4
    data 5
    repository 4
    utilities 6
Dataexp 6, 36, 43, 55, 66, 67, 68, 71, 76, 80
Dataimp 6, 36, 55, 67, 69, 71, 76
DB2
  buffer pools 14
  databases 19
  DSNUTILS 15
  DSNZPARM 14
  EDM pool 14
  indexes 24
  RUNSTATS 15
  storage groups 14
  stored procedures 15
  table spaces 19
  tables 24, 33
    source 33, 56, 67
    target 34, 57, 68
    temporary database 158
temporary table spaces 158
DB2 family cross-loader 50
DB2 for NT Export 46
DB2 for NT Import 46
DB2 LOAD 47, 83
DB2 Load 80
DSNTIAUL 47

E
EIM 38, 70
  .ifb files 41
  Export 39, 42, 71
  Import 45, 71
  interface tables 39
  relationship tables 40
  server task 41
Enterprise Integration Manager 6

M
maintenance 13
migration 33
  from DB2 for NT 31
  from Oracle for NT 53, 73
  methodology 7
  one-phase 75
  scenarios 8
  to DB2 for NT 53
  to DB2 for OS/390 31, 73
two-phase 74

O
object names
  consolidating 77, 78
  verifying 76
OS/390 environment 11
  RRS 15
  WLM 15

P
performance 3
R
RACF group ID    13
  SSEEIM   13
  SSEROLE  13
RACF user ID    13
  SADMIN   13
  SIEBEL   13
Redbooks Web Site 208
  Contact us xii
Relational Connect 83
repository 4, 27, 60, 61, 71

S
scalability 3
schema 19, 61
  changes 55
    no changes 67
server consolidation 3
Siebel 2000 setup 18
Siebel Call Center 29
Siebel Database Server 18
Siebel File System 18, 28
Siebel Gateway Server 18
Siebel Server 18
SRF file 64, 71

T
timings 50
tools 6

V
verifying object names 76
VTOC 14
Migrating Siebel Database from DB2/Oracle for NT to DB2 for OS/390
Migrating Siebel Database from DB2/Oracle for NT to DB2 for OS/390

This IBM Redbook will help you migrate the Siebel 2000 V6.2.1 database from:
- DB2 for NT V6 to DB2 for OS/390 V6
- Oracle for NT V8.1.6 to DB2 for NT V6
- Oracle for NT V8.1.6 to DB2 for OS/390 V6

The book describes the migration experiences gained while migrating a Siebel 2000 database at the IBM ITSO Poughkeepsie Center in New York, and at the IBM Hursley Laboratory in the UK.

It provides an overview of Siebel architecture, and introduces the migration methodology needed to move the Siebel database from a DB2/Oracle for NT platform to a DB2 for OS/390 platform. It offers a step-by-step description of the database migration process, and discusses in detail the different methods of moving data from one platform to the other.

This redbook will be especially useful for those migrating the Siebel database for the first time.

For more information: ibm.com/redbooks