

DATABASE CONVERSION GUIDE

FOXPRO 2.6 TO ORACLE7

VERSION 1.1
NOVEMBER 1995

Part Number C10431

ORACLE[®]

Worldwide Alliances • Design & Migration Services

Database Conversion Guide: FoxPro 2.6 to Oracle7
Version 1.1
November 1995
Part Number C10431

Contributors: Sanjoy Mondal, Shailendra Pradhan, Geeta Deodhar, Dan Mohler

Copyright © Oracle Corporation 1995

All rights reserved. Printed in the U.S.A.

This software/documentation contains proprietary information of Oracle Corporation; it is provided under a license agreement containing restrictions on use and disclosure and is also protected by copyright law. Reverse engineering of the software is prohibited. If this software /documentation is delivered to a U.S. Government Agency of the Department of Defense, then it is delivered with Restricted Rights and the following legend is applicable:

Restricted Rights Legend

Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of DFARS 252.227-7013, Rights in Technical Data and Computer Software (October 1988).

Oracle Corporation, 500 Oracle Parkway, Redwood Shores, CA 94065

If this software/documentation is delivered to a U.S. Government Agency not within the Department of Defense, then it is delivered with "Restricted Rights," as defined in FAR 52.227-14, Rights in Data-General, including Alternate III (June 1987).

The information in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing as indicated in the Preface of this document. Oracle Corporation does not warrant that this document is error-free.

This material or any portion of it may not be copied in any form or by any means without the express prior written consent of the Oracle Education Group of Oracle Corporation. Any other copying is a violation of copyright laws and may result in civil and/or criminal penalties.

Oracle, SQL*Loader, and SQL*Plus are registered trademarks of Oracle Corporation. Oracle7 and PL/SQL are trademarks of Oracle Corporation.

All products or company names are used for identification purposes only, and may be trademarks of their respective owners, specifically including Microsoft Corporation.



PREFACE

This guide explains how to convert a FoxPro 2.6 database to an Oracle7 database. It includes utilities for automating the conversion of both an application database and the data within it.

Audience

The information in this manual is intended primarily for Oracle Value Added Resellers (VARs) and Oracle consultants. Readers should have a working knowledge of SQL, PL/SQL, SQL*Plus and Oracle concepts.

How This Guide Is Organized

Chapter 1: Overview

This chapter provides an overview of converting a FoxPro 2.6 database to an Oracle7 database.

Chapter 2: Performing the Conversion

This chapter describes the procedures used to convert the source FoxPro 2.6 database to the target Oracle7 database.

Chapter 3: Datatypes

This chapter provides detailed descriptions of the differences in data types used by the FoxPro 2.6 and Oracle7 databases.

Chapter 4: Comparing FoxPro 2.6 and Oracle7

This chapter discusses the differences between FoxPro 2.6 and Oracle7.

Chapter 5: Data Storage Concepts

This chapter provides a detailed descriptions of the conceptual differences in data storage for the FoxPro 2.6 and Oracle7 databases.

Appendix A: Oracle7 Normalized Schema

This appendix covers the Oracle7 schema for the normalized tables used in the intermediate Oracle7 database.

Appendix B: System Tables

This appendix covers the Oracle7 and FoxPro 2.6 system tables.

Appendix C: Conversion Estimation Guide

This appendix helps you estimate the time and processes involved in the conversion.

Related Publications

Along with this guide, you may want to refer to the following documents published by Oracle Corporation:

- *ORACLE7 Server Concepts Manual*, Part No. 6693-70
- *ORACLE7 Server Administrator's Guide*, Part No. 6694-70
- *ORACLE7 Server Application Developer's Guide*, Part No. 6695-70
- *ORACLE7 Server Messages and Codes Manual*, Part No. 3605-70
- *ORACLE7 Server Parallel Server Administrator's Guide*, Part No. 5990-70

- *ORACLE7 Server SQL Language Reference Manual*, Version 7, Part No. 778-70
- *PL/SQL User's Guide and Reference*, Part No. 800-V1.0
- *Programmer's Guide to ORACLE Precompilers*, Part No. 5315-15
- *Pro*C Precompiler Supplement*, Part No. 5452-15
- *Pro*COBOL Precompiler Supplement*, Part No. 5451-15
- *Pro*FORTRAN Precompiler Supplement*, Part No. 5453-15
- *Pro*Pascal Precompiler Supplement*, Part No. 5455-15
- *Pro*PL/I Precompiler Supplement*, Part No. 5454-15
- *SQL*Net Administrator's Guide*, Part No. A11325-1

References

You may want to refer to FoxPro reference materials for specific details on FoxPro functionality.

Conventions Used in This Guide

The following conventions are used in this guide:

UPPERCASE	Calls attention to command keywords, command names, table names, object names, and filenames. Enter text exactly as spelled; it need not be in uppercase.
<i>lowercase, italics</i>	A clause value; substitute an appropriate value.
font change	A different font distinguishes examples of commands and statements from the rest of the text.
bold words	Calls attention to important information.
	Separates alternative syntax items that may be optional or mandatory. Do not enter the .
[]	One or more optional items. Do not enter the brackets.
{ }	A choice of mandatory items; enter one of the items. Do not enter the braces.
. . .	Preceding item(s) may be repeated any number of times.

Your Comments Are Valued

We welcome and appreciate your comments. As we develop our software and documentation, your opinions are the most important input we receive. To contact us, send your e-mail to:

`infockit@us.oracle.com`

If you prefer, call us at (415) 506-0329 or write to:

Oracle Corporation
World Headquarters
Worldwide Alliances
Design & Migration Services
Box 659604
500 Oracle Parkway
Redwood Shores, California 94065
U.S.A.



CONTENTS

Chapter 1	Overview	1-1
Introduction to the Conversion		1-2
Prerequisites for Using the Conversion Kit		1-2
Descriptions of the Conversion Kit Utilities.....		1-3
How the Schema Mover Works.....		1-3
How the Data Mover Works.....		1-4
Capabilities of the Conversion Kit Utilities.....		1-5
Schema Mover Capabilities.....		1-5
Chapter 2	Performing the Conversion	2-1
Overview of the Conversion Process.....		2-2
Moving the Database Schema Information.....		2-2
Moving the Database Application Data.....		2-2
Graphic Overview of the Conversion Process		2-3
Getting Started.....		2-4
FoxPro Conversion Kit Files		2-4
Oracle7 Conversion Kit Files		2-5
Moving the Database Schema.....		2-7
Capturing the Database Schema Information		2-7
Handling Special Conditions in the Conversion.....		2-9
Creating the Target Oracle7 Database.....		2-10
Moving the Database Data.....		2-12

Extracting the Application Data	2-12
Loading the Application Data.....	2-13
Chapter 3.....	Datatypes 3-1
Chapter 4.....	Comparing FoxPro 2.6 and Oracle7 4-1
FoxPro 2.6 File Manager vs. Oracle7 Full-Featured RDBMS.....	4-2
Data Dictionary Support	4-4
Benefits of Normalized Database Design	4-5
SQL Interface	4-7
Chapter 5.....	Data Storage Concepts 5-1
The FoxPro 2.6 File/Table vs. the Oracle7 Table.....	5-2
FoxPro 2.6 Key and Index vs. Oracle7 Key, Index, and Constraint	5-4
FoxPro 2.6 Database vs. the Oracle7 Database	5-5
Appendix A.....	Oracle7 Normalized Schema A-1
Appendix B.....	System Tables B-1
Appendix C.....	Conversion Estimation Guide C-1

1

OVERVIEW

This chapter provides an overview of converting a FoxPro 2.6 database to an Oracle7 database. Specifically, this chapter contains the following information:

- a high-level introduction to the conversion including a list of prerequisites that are required for you to use the conversion kit
- general descriptions of how the conversion kit utilities work
- detailed descriptions of the capabilities of the conversion kit utilities

Introduction to the Conversion

Conversion from one database to another can be a complex operation. This FoxPro 2.6 to Oracle7 conversion kit helps you start the process of converting the data and applications from a FoxPro 2.6 database to an Oracle7 database and provides a reference source for use throughout the conversion. The kit is for value-added resellers (VARs) and Oracle consultants who need to convert FoxPro 2.6 database applications to run on an Oracle7 database.

Converting database applications can be divided into two high-level parts. The first part is to convert the database and the second is to convert the applications that run on the database. Since you need to convert the database before converting the applications, the primary focus of this initial version of the conversion kit is on supplying software utilities and documentation that will help you to quickly complete the first part of the conversion, the conversion of the database.

Prerequisites for Using the Conversion Kit

Before you can use the conversion kit, you need to:

1. Have the following required database software installed and running on your computer system:
 - Microsoft FoxPro 2.6 software
 - FoxPro 2.6 for Windows
 - Oracle7 database software
 - Oracle7 with the Procedural Option
 - SQL*Plus
 - SQL*Loader
2. Initially setup the Oracle7 database as follows:
 - Create an empty Oracle7 database.
 - Create one or more TABLESPACES (logical data storage areas) in the Oracle7 database.
3. Understand how to use the FoxPro 2.6 and Oracle7 tools listed above.

Description of the Conversion Kit Utilities

The task of converting the database is handled by the two sets of utility programs supplied with the conversion kit. The first set of utilities function as a schema mover, which converts the FoxPro 2.6 database schema definitions into Oracle7 database schema definitions. The second set functions as a data mover, which converts and moves the application data from the source FoxPro 2.6 database to a target Oracle7 database.

How the Schema Mover Works

The schema mover utilities convert the schema information from the source FoxPro 2.6 database into the database schema for the target Oracle7 database.

The schema mover is divided into the following three functional parts:

1. Capturing the source FoxPro 2.6 database schema information. This part performs the following functions during the conversion:
 - provides instructions for preparing the source FoxPro 2.6 database for conversion
 - uses FoxPro 2.6 database utilities to export the schema information from the source FoxPro 2.6 database to flat files
 - uses the Oracle SQL*Loader utility to upload the schema information from the flat files into a set of normalized tables in an intermediate Oracle7 database

The intermediate Oracle7 database is a temporary repository that stores the schema information in a special format defined for use by the conversion kit. This repository is implemented by defining a set of tables in an Oracle7 database. The conversion kit provides you with scripts to create these tables. You can delete this intermediate database after successfully completing the conversion.

2. Manually handling special conditions in the conversion. In some of these conditions you have the opportunity to override a default chosen by the schema mover. In others, the schema mover can not find an acceptable default and requires that you enter one.
3. Creating the target Oracle7 database. This part performs the following functions during the conversion:
 - uses PL/SQL procedures to create DDL scripts from the normalized tables in the intermediate Oracle7 database
 - executes the DDL scripts to create the target Oracle7 database

The target Oracle7 database is a functional Oracle7 database that is ready to accept data after the schema mover parts of the conversion are complete. The data mover part of the conversion transfers the application data to the target database, as described below.

How the Data Mover Works

The conversion kit first uses the schema mover utilities to convert the FoxPro 2.6 database schema information into the Oracle7 database schema. The data mover utilities then convert and move the application data from the FoxPro 2.6 database to the Oracle7 database. The conversion kit uses FoxPro 2.6 utilities to download the data from the FoxPro 2.6 database to flat files, which are used to load the data into the Oracle7 database.

The data mover is divided into the following two functional parts:

1. Using FoxPro 2.6 database utilities to automatically export all of the application data from the source FoxPro 2.6 database to flat files
2. Using the Oracle SQL*Loader utility to load the data into the target Oracle7 database

The target Oracle7 database is a functional database that contains the data from the source FoxPro 2.6 database after you complete the data mover steps of the conversion.

Capabilities of the Conversion Kit Utilities

The schema mover and data mover utility programs automate most parts of moving the schema information and application data from the source FoxPro 2.6 database to the target Oracle7 database. Due to differences in the features of the FoxPro 2.6 and Oracle7 databases and in the implementation of similar database functions, some parts of the conversion require manual input and some items cannot be directly converted. You should familiarize yourself with these differences before you begin the conversion. Handling of these conditions is discussed further in Chapter 2.

Schema Mover Capabilities

The capabilities of the schema mover are described below according to the following levels:

- fully automated conversion
- automated conversion functions requiring user input
- manual conversion as described in this guide and other Oracle Corporation manuals, but not automated

Fully Automated Conversion Functions

The following conversion functions are fully automated by utility programs supplied with the conversion kit:

- Information is automatically downloaded from the source FoxPro 2.6 database including the following:
 - columns
 - tables

Automated Conversion Functions Requiring User Input

The following conversion functions are automated by the utility programs but require your manual input during the conversion:

- To convert FoxPro 2.6 indexes to Oracle7 indexes, you need to input the index filename and table name for the index to be created.
- Objects in the source FoxPro 2.6 database with a name that is a reserved word in Oracle7 must be renamed.

The conversion utilities identify FoxPro 2.6 database object names that are reserved words in Oracle7. For a list of Oracle7 database reserved words, see the *Oracle7 Server SQL Language Reference Manual* and the *PL/SQL User's Guide and Reference*.

- Default information must be input for passwords and tablespaces because defaults are required for Oracle7 databases but are not used by FoxPro 2.6 databases.

Manual Conversion Functions

The following conversion functions are not automated by the conversion utilities and require you to make manual changes in the Oracle7 database after the conversion.

- FoxPro 2.6 does not support granting privileges on any of the objects to the users. FoxPro 2.6 provides no security features. You can implement security features with Oracle7 by creating users and granting privileges on different objects to different users. The conversion utility creates only one user ID, Oracle, that owns all the objects after the conversion. You can create more users and grant privileges on objects to other users.
- If there are two or more MEMO/GENERAL columns in a FoxPro 2.6 table, this table must be recreated in the Oracle7 database with only one LONG column. You may create more tables to have other LONG columns. Note that MEMO columns are converted to LONG columns and GENERAL columns are converted to LONG RAW columns.

2

PERFORMING THE CONVERSION

This chapter describes the procedures used to convert your source FoxPro 2.6 database to an Oracle7 database. Specifically, this chapter contains the following information:

- a list of step-by-step procedures you use to install the conversion kit software
- lists of step-by-step procedures you use to convert the database schema information
- lists of step-by-step procedures you use to convert the application data

Overview of the Conversion Process

This overview of the process of converting a source FoxPro 2.6 database to a target Oracle7 database will assist you in following the step-by-step conversion procedures given in this chapter. The conversion process consists of using the schema mover utilities to move the database schema information and the data mover utilities to move the database application data. Moving the schema information is described in three functional parts. Moving the application data is described in two functional parts. Each part has one or more sets of step-by-step instructions.

Moving the Database Schema Information

The process of moving the schema information from the source FoxPro 2.6 database to the target Oracle7 database consists of three functional parts.

Part one of the schema mover is described in a series of numbered steps for procedures that use the conversion kit utilities to capture the source FoxPro 2.6 database schema information.

Capturing the source database schema information includes a set of steps each for:

1. Preparing the source FoxPro 2.6 database schema information.
2. Extracting the schema information from the FoxPro 2.6 database.
3. Loading the schema information into an intermediate Oracle7 database.

Part two of the schema mover is described in a series of numbered points to follow in handling special conditions in the conversion that can not be automated by the schema mover utilities.

Part three of the schema mover is described in a series of numbered steps for procedures that use the conversion kit utilities to create the target Oracle7 database.

Creating the target Oracle7 database includes a set of steps each for:

1. Creating Oracle7 DDL statements.
2. Executing the DDL scripts to create the target Oracle7 database.

Moving the Database Application Data

The process of moving the application data from the source FoxPro 2.6 database to the target Oracle7 database consists of two functional parts.

Part one of the data mover is described in a series of numbered steps for procedures that use the conversion kit utilities to retrieve the application data from the source FoxPro 2.6 database.

Part two of the data mover is described in a series of numbered steps for procedures that use the conversion kit utilities to load the application data into the target Oracle7 database.

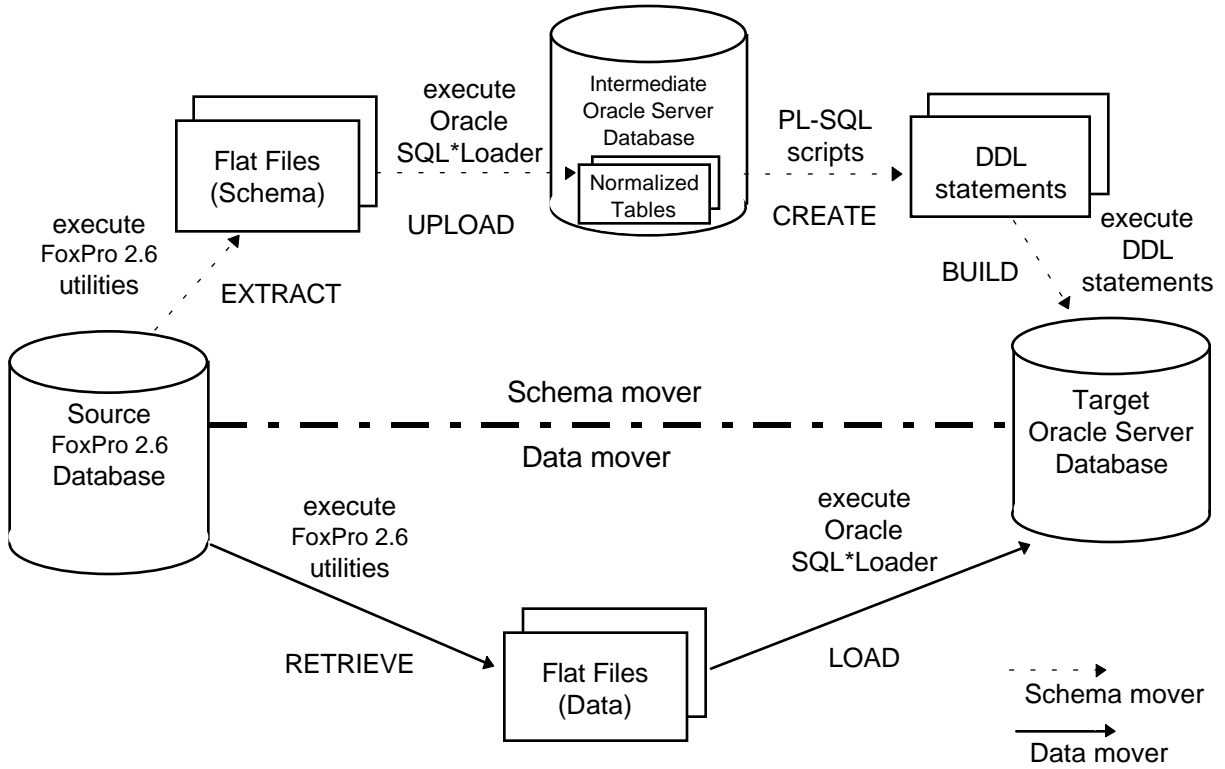


Figure 1. Overview of the Conversion Process

Getting Started

Before you can begin the database conversion process, you must prepare the systems that you will be using to perform the conversion.

To prepare for the conversion:

1. Copy the files from the `\convkit\foxpro26` directory on the provided media to a directory on a system that has access to the source FoxPro 2.6 database.
2. Copy the files from the `\convkit\oracle` directory on the provided media to a directory on a system that has access to the target Oracle7 database.
3. Use the file lists below to verify that the conversion kit files install successfully.

FoxPro 2.6 Conversion Kit Files

The following files should exist on the system with the FoxPro 2.6 database:

FoxPro 2.6 Conversion Kit Files under `\convkit\foxpro26\`

CONVSD.SCX	User interface screen source code
CONVSD.SCT	Intermediate file generated for user interface screen file convsd.scx
CONVSD.SPR	File generated from user interface screen source code
CONVSD.SPX	Intermediate file generated from user interface screen source code
CRSCHSQL.PRG	Script to create F2O_INDIX table.
DATACONV.PRG	FoxPro 2.6 source file for data conversion
DATACONV.FXP	FoxPro 2.6 compatible object file generated from datconv.prg
HELP.BMP	Image file for help icon
LDIDXSQL.PRG	An example script to load F2O_INDIX table.
SCHCONV.PRG	FoxPro 2.6 source file schema conversion
SCHCONV.FXP	FoxPro 2.6 compatible object file generated from schconv.prg
W2BACK.BMP	An image file for icon
W2END.BMP	An image file for icon
W2NEXT.BMP	An image file for icon
W2TOP.BMP	An image file for icon

Oracle7 Conversion Kit Files

The following files should exist on the system with the Oracle7 database:

Oracle7 Conversion Kit Files under \convkit\oracle\schema\

CONVRS.SQL	Creates a rollback segment called CONVERSION.
CONVTS.SQL	Creates a tablespace CONVERSION.
CONVUSER.SQL	Creates user CONVUSER for use by the conversion utility.
CRTABLES.SQL	Creates normalized tables required for an intermediate Oracle7 database.
FINDRESV.SQL	Generates a report containing reserved words used in user name, table name, column name, and index name. The report file name is RESVWORD.REP.
GNIDXDDL.SQL	Spools DDL for index creation in INDEX.DDL.
GNTBLDDL.SQL	Spools DDL for table creation in TBL.DDL.
GNUSRDDL.SQL	Spools DDL for user creation in USER.DDL.
INDEX.SCR	Generates CREATE INDEX statements for indexes attached to tables.
INS_RESW.SQL	Loads Oracle7 reserved words into intermediate Oracle7 database.
TBL.SCR	Generates CREATE TABLE statements for tables.
USER.SCR	Generates CREATE USER statements for users.

Oracle7 Conversion Kit Files under \convkit\oracle\schema\fp_ora\

INS_DTYP.SQL	SQL script to load datatype information.
INS_ROLE.SQL	SQL script to create role, granting privileges to role and assigning role to user
LDSHEMA.BAT	Script consisting of SQL*Loader commands to load schema information.
OBJ.CTL	SQL*Loader control file to load object information in intermediate Oracle7 database.
TBL.CTL	SQL*Loader control file to load table information in intermediate Oracle7 database.
TBL_COL.CTL	SQL*Loader control file to load table column information in intermediate Oracle7 database.
USER.CTL	SQL*Loader control file to load user information in intermediate Oracle7 database.

Oracle7 Conversion Kit Files under \convkit\oracle\data\fp_ora\

CRCTL.SQL	Creates control files. (This is an intermediate file created by the utilities.)
GENCTL.SCR	Generates SQL*Loader control files for every table.
GENLIST.SQL	Spools the commands to generate a SQL*Loader control file for each table into the file CRCTRL.SQL.
GENSCRIPT.SQL	Spools SQL*Loader commands into LOADDATA.BAT.
LOADDATA.BAT	Batch file for calling SQL*Loader. (This is an intermediate file created by the utilities.)

4. If your Oracle7 database only has one rollback segment, add another one with the script below.

```
sqlplus system/manager
@convkit\oracle\schema\convrs.sql
```

5. Create a separate tablespace for use by the conversion kit. The tables created by the conversion utilities are all placed on this tablespace. You need to modify the script below to specify the correct file name for this tablespace. If you decide to use a different tablespace than the one defined in the script below, change the convuser.sql script to specify the correct default tablespace.

```
sqlplus system/manager
@convkit\oracle\schema\convts.sql
```

6. Create a user account on your target Oracle7 database for use by the conversion utility. You may use any username and password by replacing convuser with your username and dms with your password within the convuser.sql script and throughout the steps of the conversion. After you have completed the conversion, you may drop this account.

```
sqlplus system/manager
@convkit\oracle\schema\convuser.sql
```

Moving the Database Schema

The schema mover converts the schema information from the source FoxPro 2.6 database into the database schema for the target Oracle7 database.

The process of moving the database schema consists of the following functional parts:

- capturing the source FoxPro 2.6 database schema information
- handling special conditions in the conversion
- creating the target Oracle7 database

Capturing the Database Schema Information

The first part of the schema mover process is capturing the source FoxPro 2.6 database schema information. This part includes:

- preparing the source FoxPro 2.6 database
- extracting the schema information from the source FoxPro 2.6 database into flat files
- loading the schema information from the flat files into a set of normalized tables in an intermediate Oracle7 database

Preparing the Source FoxPro 2.6 Database

- Put all the .DBF files that you want to convert into a directory.
- Put all the index files (.IDX and .CDX) that you want to convert into the directory where you have put all the .DBF files.
- Modify the script LDIDXSQL.PRG in the convkit\foxpro26 directory to insert the values for F2O_INDX table. The first value is the table name. The second value is the index name, and the third value is the type of index—IDX or CDX. Insert one row for each and every index.

Extracting the Schema Information from the FoxPro 2.6 Database

Use the following steps to extract the schema information from the source FoxPro 2.6 database to flat files.

These steps extract the information by executing a screen, convsd.scx. Information from the .DBF files are unloaded into flat files.

1. Change into the \convkit\foxpro26 directory on the system with the source FoxPro 2.6 database.
2. Double-click on the FoxPro icon to start FoxPro 2.6.
3. On the Menu bar, click on Program.
4. Click on Do. A pop-up window appears.

5. Double-click on the c:\convkit\foxpro26 directory, and select the CRSCHSQL.PRG file.
6. Click the OK button.
7. Double-click on the c:\convkit\foxpro26 directory, and select the LDIDXSQL.PRG file.
8. Click on the OK button.
9. Double-click on the c:\convkit\foxpro26 directory, and select the CONVSD.SPR file.
10. Click on the OK button.
11. Either type in the name of the source directory where you have stored the .DBF files that you want to convert, or click on the Browse button to find the source directory as shown in Figure 2.

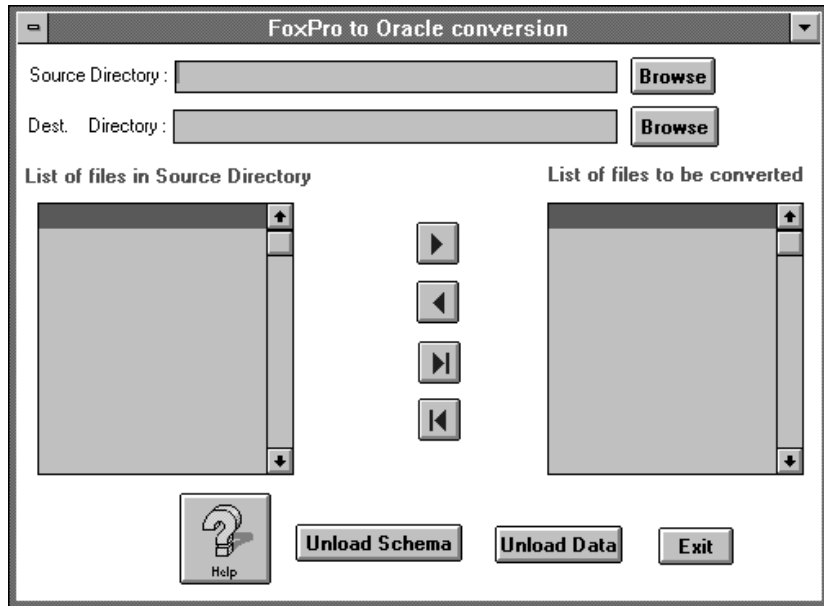


Figure 2. Selecting the Source and Destination Directories

12. Select the files that you want to convert, or select them all. You can also remove files from the list of files to be converted.
13. Either type in the name of the destination directory where you want to store the unloaded schema information, or click on the Browse button to find the destination directory as shown in Figure 2.
14. Click on Unload Schema. This will unload the schema information from the .DBF files into a number of files with the extension .DAT.
15. Copy the .DAT files to the system that has access to the Oracle7 database. Put the files in the \convkit\oracle\schema\fp_ora directory.

Loading the Schema Information into the Intermediate Oracle7 Database

Use the following steps to load the schema information from the flat files created in the steps listed above into normalized tables in the intermediate Oracle7 database.

1. Change into the `\convkit\oracle\schema` directory on the system with the Oracle7 Server database.
2. Execute the `crtables.sql` script to create the normalized tables required for the intermediate Oracle7 database. After the script completes, check the file `crtables.out` for any errors. Ignore “ORA-942 Table or View does not exist” errors. These are normal.

```
sqlplus convuser/dms @crtables.sql
```

3. Execute the `ins_dtyp.sql` script to insert the data type mapping into the `DATATYPE_INFO` table. After the script completes, check the file `ins_dtyp.out` for any errors.

```
sqlplus convuser/dms @fp_ora\ins_dtyp.sql
```

4. Execute the `ins_resw.sql` script to insert the Oracle7 reserved words into the `ORA_RESERVED_WORDS` table. After the script completes, check the file `ins_resw.out` for any errors.

```
sqlplus convuser/dms @ins_resw.sql
```

5. Change into the `\convkit\oracle\schema\fp_ora` directory.
6. Execute the `ldschem.bat` script to load the source FoxPro 2.6 database schema information from the `.dat` files into the normalized tables in the intermediate Oracle7 database. If you are not using `convuser/dms` for the username and password, you must edit this script to include your username and password.

The `ldschem.bat` is a batch file that executes `SQL*Loader`.

7. Change into the `\convkit\oracle\schema` directory.

Handling Special Conditions in the Conversion

Now that you have captured the source FoxPro 2.6 schema information, you are ready to begin the second part of the schema mover process. You now need to handle special conditions in the conversion.

There are two types of special conditions. The first set below are conditions in which your manual intervention is required. The second set explains conditions in which the schema mover chooses a default unless you make a modification.

Conditions requiring manual modification:

1. Reserved words used in object names.

There are many reserved words in Oracle7 which are valid object or column names in SQL Server. Execute the `findresv.sql` script to identify names

which are reserved words. This script will create a file called `resvword.rep`. Check this report for any reserved word usage in your schema.

```
sqlplus convuser/dms @findresv.sql
```

Change the conflicting names in the intermediate database.

Conditions allowing you to override a default:

1. Default user information.

Oracle7 Server databases allow you to define a number of defaults when a database user is created.

Only one user is created by the conversion utilities. You can create more users after the conversion is complete. For each user created in the target Oracle7 database, there is a record in the `USER_INFO` table. Update the columns `DEF_TBLSP`, `REAL_PASSWORD`, `TEMP_TBLSP` and `PROFILE_NAME` with appropriate values to take advantage of this additional control. If you do not enter values for these columns, the database chooses a default automatically. The default `DEF_TBLSP` is `SYSTEM`, which is almost always a bad choice for user data.

Creating the Target Oracle7 Database

Now that you captured the source FoxPro 2.6 database schema information and handled the special conditions, you are ready to begin the third part of the schema mover process.

The third part of the schema mover process is creating the target Oracle7 database.

This part includes:

- using PL/SQL procedures to create DDL scripts from the normalized tables in the intermediate Oracle7 database
- executing the DDL scripts to build the target Oracle7 database schema information

Using PL/SQL Scripts to Create DDL Statements

To begin, change into the `convkit\oracle\schema` directory. Then execute the following PL/SQL scripts. After each script, you can check for errors in files called `<filename>.out` (e.g., `user.out`).

NOTE: If you replaced `convuser` with your username and `dms` with your password, be sure to make the replacements before executing these scripts.

1. `sqlplus convuser/dms @user.scr`
2. `sqlplus convuser/dms @tbl.scr`
3. `sqlplus convuser/dms @index.scr`

Executing the DDL Scripts to Create the Target Oracle7 Database

Now that you have created the DDL scripts, execute them in the following order to create the target Oracle7 database:

1. `sqlplus system/manager @user.ddl`
2. `sqlplus system/manager @fp_ora\ins_role.sql`
3. `sqlplus system/manager @tbl.ddl`
4. `sqlplus system/manager @index.ddl`

This process completes the creation of the required DDL.

Moving the Database Data

The data mover converts and moves the application data from the source FoxPro 2.6 database to the target Oracle7 database.

Moving the application data consists of the following two functional parts:

- extracting the application data from the source FoxPro 2.6 database to flat files by using FoxPro 2.6 database utilities
- converting and moving the data from the source FoxPro 2.6 database to the target Oracle7 database by using the Oracle7 database SQL*Loader utility.

Extracting the Application Data from the Source FoxPro 2.6 Database

Perform the following steps to extract the application data from the source FoxPro 2.6 database into flat files:

1. Change into the `\convkit\foxpro26\data` directory on the system with the FoxPro 2.6 database.
2. Double-click on the FoxPro icon to start FoxPro 2.6.
3. On the Menu bar, click on Program.
4. Click on Do. A pop-up window appears.
5. Double-click on the `c:\convkit\foxpro26` directory, and select the `convsd.spr` file.
6. Click on the OK button.
7. Either type in the name of the source directory where you have stored the .DBF files that you want to convert, or click on the Browse button to find the source directory. (Refer to Figure 2 on page 2-8.)
8. Select the files that you want to convert, or select them all. You can also remove files from the list of files to be converted.
9. Either type in the name of the destination directory where you want to store the converted files, or click on the Browse button to find the destination directory. (Refer to Figure 2 on page 2-8.)
10. Click on Unload Data. This creates a set of ASCII text files with the data. Each .DBF file is loaded into one flat file. The names of the flat files are generated using the .DBF file name. For example, the file with `customer.dbf` will get loaded into the flat file named `customer.dat`.
11. Copy the data files to a system that has access to the target Oracle7 database. Put the files in `\convkit\oracle\data\fp_ora`.

Loading the Application Data into the Target Oracle7 Database

Perform the following steps to load the application data from the flat files created in the previous set of steps into the target Oracle7 database:

1. Change into the `\convkit\oracle\data\fp_ora` directory on the system with the Oracle7 database.
2. Execute the `genctl.scr` script:

```
sqlplus convuser/dms @genctl.scr
```

This creates the SQL*Loader control files and a batch file named `loaddata.bat` which you use to load the data. After the script completes, check the file `genctl.out` for any errors.

3. Execute the `loaddata.bat` script to load the application data from the flat files into the target Oracle7 database.

You have now completed the steps for schema and data moving. The next chapter provides reference information on Datatypes to assist you in converting your database environment to Oracle.

3

DATATYPES

This chapter provides detailed descriptions of the differences in datatypes used by theFoxPro 2.6 and Oracle7 databases. Specifically, this chapter contains the following information:

- a table showing the base FoxPro 2.6 datatypes available and how they are mapped to Oracle7 datatypes
- recommendations based on the information listed in the table.

Datatypes Table

FoxPro 2.6	Description	Oracle7	Comments
NUMERIC(p,[s]) FLOAT(p,s)	Stores decimal floating point numbers up to a maximum of 20 significant digits. <i>p</i> is the number of significant digits and <i>s</i> is the number of digits to the right of the decimal point.	NUMBER(p,[s])	
LOGICAL	Data is stored as T (True) or F (False).	NUMBER(1)	True can be converted to 1, False can be converted to 0.
CHAR(n), CHARACTER(n)	A fixed length string of exactly <i>n</i> characters, blank padded. It requires 1 byte per character. $0 < n < 255$	CHAR(n)	
MEMO	Stores any kind of text data. One table can contain one or more MEMO columns.	LONG	The LONG field can store character data of variable length up to 2 gigabytes. One table can contain only one LONG column.
GENERAL	Stores any kind of binary data in an undifferentiated stream. One table can contain one or more GENERAL columns.	LONG RAW	The LONG RAW field can store binary data of variable length up to 2 gigabytes. One table can contain only one LONG RAW column.
DATE	Default format is mm/dd/yr.	DATE	Default format is dd-mon-yr.

Recommendations:

The datatypes can easily be converted from FoxPro 2.6 to Oracle7 with the equivalent datatypes listed in the above table.

4

COMPARING FOXPRO 2.6 AND ORACLE7

To successfully migrate applications and data from FoxPro 2.6 to Oracle7, you must first understand the differences between the two environments.

FoxPro 2.6 and Oracle7 are very different in form and function. This chapter provides a feature-by-feature comparison of the two systems so you can compensate for the differences when migrating databases and applications from FoxPro 2.6 to Oracle7.

The major differences between FoxPro 2.6 and Oracle7 fall into the following categories:

Category	FoxPro 2.6	Oracle7
Product Type	File manager/ relational	Relational database
Data dictionary	No	Complete
Normalization	Optional	Standard
Access Method	4GL, ODBC	3GL (C & Pro*C)/ 4GL/ ANSI compliant SQL interface

FoxPro 2.6 File Manager vs. Oracle7 Full-Featured RDBMS

In FoxPro 2.6, there is a flat file for each and every table. To open any table, you have to 'USE' that table, and then you can manipulate that table. Locking facilities are provided through calls, and the calling application program is responsible for locking tables.

Oracle7, in contrast, is a full-featured database offering:

- declarative integrity
- stored procedures
- parallel query
- replication
- SQL & PL/SQL (at both client and server)
- on-line backup
- distributed transactions

Integrity Protection

Oracle7 provides extensive data integrity protection at the database level rather than at the application level. So, regardless of where changes come from, the database is guaranteed to remain consistent. New application programs immediately benefit from this functionality because they do not need to repeat functions and integrity constraints for the database.

Administration

As of Oracle7 Release 7.3, the workgroup manager will be provided to enable remote administration of multiple databases. FoxPro 2.6 provides utilities to debug applications at both compile time and run time. Oracle7 provides an analogous utility in its SQL*DBA program and its successor, Server Manager.

Index Differences

In FoxPro 2.6, you have three types of indexes—simple, structural compound, and non-structural compound indexes. If you want to use any index, you have to specify the index name when you use the table.

Oracle7 does not need any index at all to find data. If no index exists for a given query, Oracle7 does a "full table scan" where it reads every record in the table. For small tables this is faster than using indexes. Beyond that, Oracle7 uses an exactly matching index to efficiently evaluate the query.

But when the query needs more index columns than are provided, Oracle7 is able to use partial indexes to restrict the number of rows it needs to examine to fulfill a query.

Example:

Table t1 consists of columns:

c1 c2 c3 c4 ...

Oracle7 index i1 on t1 consists of one field only:

c1

Oracle7 is able to use i1 to retrieve data for any queries of this type:

c1,
c1 || c2,
c1 || c2 || c3, ...

FoxPro 2.6 would require each of these combinations to be individually indexed. Thus, depending on the performance requirements of an application, indexes in Oracle7 should be added far more conservatively than under FoxPro 2.6. This leaner system saves you disk space and improves insertion and deletion performance as well.

Data Dictionary Support

FoxPro 2.6 maintains one flat file per table. That flat file contains table structure and data information. You cannot query these tables using simple SQL statements.

Oracle7, by contrast, maintains a complete *data dictionary*: a set of read-only reference tables and views containing complete descriptive information on all database objects.

The data dictionary describes in detail:

- the names of Oracle7 users
- privileges and roles each user has been granted
- names and attributes of schema objects (tables, views, snapshots, indexes, clusters, synonyms, sequences, procedures, functions, packages, triggers, etc.)
- information about integrity constraints
- default values for columns
- the amount space allocated for, and currently used by, the objects in a database
- auditing information, such as who has accessed or updated various objects
- other general database information
- referential integrity constraints

Benefits of Normalized Database Design

For relational or non-relational data management systems, there are several benefits of normalized database design, including:

- better data integrity
- more flexible data access
- reduced disk space consumption

Oracle7 offers powerful built-in features, such as declarative referential integrity constraints, that complement normalized database design. Therefore, when converting from FoxPro 2.6 to Oracle7, we suggest you review your database design and strongly consider increasing its degree of normalization.

In general, to normalize your database, you should establish unique keys for all rows. FoxPro 2.6 allows equal-valued (non-unique) keys which can cause complications following your database conversion. By default, Oracle7 does not distinguish between equal-valued keys, so there can be a problem in identifying rows for processing: a programmer has difficulty determining the correct row among the set retrieved by a query.

FoxPro 2.6 gets around this by having certain functions that use logical record number.

This feature is also available under Oracle7 in the form of a semi-permanent ROWID. The ROWID is guaranteed to remain constant for the duration of a transaction only. In practice, it rarely changes, but over time it will change as tables reorganize. Thus the ROWID is useful in artificially forcing a distinction between duplicate key values within a single transaction.

A much better solution is simply to ensure that a unique key exists for each row. Then the desired ordering can be obtained via the ORDER BY clause.

This conversion kit helps you to convert a FoxPro 2.6 database to a logically equivalent Oracle7 database, without making database design changes such as normalization. Database design changes involve complex cost vs. benefit tradeoffs that are beyond the scope of this paper. After using our kit to complete your database conversion, you may want to retain a relational database design expert to review your database design and application code, and help you to decide if changes would improve your system performance.

Relational Set-oriented Data Access

FoxPro 2.6 provides SQL constructs as well as FoxPro 2.6 specific calls to communicate to the database. The tables referred to in the SQL construct need to be opened before a SQL statement can be executed.

You use the SQL language to communicate with Oracle7 databases. In contrast to FoxPro 2.6, Oracle's SQL is compliant with SQL standards, so most

documentation on the subject is useful to new programmers. PL/SQL is Oracle's proprietary procedural extension to SQL.

SQL specifies what data (set of rows) to operate on. The relational database picks the optimal data access strategy at statement execution time and performs the desired action. The strategy chosen depends upon the state of the database, physical data distribution, and the existence of other database objects such as indexes that may speed access.

The set retrieved by SQL can consist of 0, 1, or any positive number of records. SQL does not order the records unless you specify an `ORDER BY` clause.

Oracle7 Cursors

SQL is not restricted by FoxPro 2.6's concept of "current record". However, PL/SQL and precompiler products can simulate this effect by using a *cursor*, a mechanism for returning one or more records at a time to a program from the result set of a query. With a cursor, there is a "current record," but it is never manipulated directly; it is an internal pointer that advances as records are read from the cursor.

Oracle's cursors are not currently "reverse scrollable"; that is, records, once read, cannot be un-read. If you need this effect, several methods are possible:

- The best way is to buffer the records in local memory and perform scrolling there.
- Failing that, records can be buffered to local disk if the performance penalty is acceptable.
- If that is still unacceptable, the result set can be stored in a temporary database table and the cursor can be quickly re-run on this smaller number of records.

In many cases, it might help you to "step back" from a strictly procedural translation in FoxPro 2.6 and use a SQL approach. For example, many programs do a sequential read of their data files. Rather than coding something like `SELECT * FROM ... WHERE ... > key`, which would have to be evaluated for each row to be retrieved, a far faster approach would be to use a cursor, which evaluates and runs the query once then queues the data for retrieval. Then, in the processing loop, one record is returned from the database at very little additional cost. This efficiently duplicates the `SCAN NEXT` behavior so often needed in sequential processing.

SQL Interface

Many FoxPro 2.6 operations do not have exact SQL equivalents. It is possible to create a one-to-one translation for most operations, but performance may not be optimal. Typically, vast portions of a FoxPro 2.6 program can be collapsed into a few SQL statements. Developers should consider every opportunity to replace laborious 3GL constructs with 4GL versions.

An application program can submit SQL statements to an Oracle7 database using one of these methods:

- language precompilers
- OCI
- PL/SQL

Language Precompilers

The Oracle precompilers are a family of products (e.g., Pro*C, Pro*FORTRAN, Pro*COBOL, etc.) that allow you to embed SQL statements in a host language program.

You should consult the *Programmer's Guide to ORACLE Precompilers* and the smaller supplement for your host language (e.g., *Pro*C Precompiler Supplement*) for more information on the capabilities of the precompilers. The Oracle precompilers also allow you to include PL/SQL code.

OCI Interface

If you need a procedural interface to the database, the Oracle Call Interface (OCI) can be used in C, FORTRAN, or COBOL programs. OCI gives you greater control and provides 5% to 20% better performance. However, it also increases coding complexity. Also, OCI is Oracle-specific whereas the precompiler programs using ANSI SQL can be ported more easily to other SQL data sources. For these reasons, we generally recommend you use precompilers as the access method.

PL/SQL

PL/SQL is Oracle's extensions to SQL allowing SQL statements to be embedded in procedural constructs. This code can be used in Oracle programming tools to improve the coding and run-time efficiency of various operations.

5

DATA STORAGE CONCEPTS

This chapter provides a detailed description of the conceptual differences in data storage for the FoxPro 2.6 and Oracle7 databases.

Specifically, this section contains tables comparing these structures:

- FoxPro 2.6 file/table vs. Oracle7 table
- FoxPro 2.6 index vs. Oracle7 key, index, and constraint
- FoxPro 2.6 database vs. Oracle7 database

The FoxPro 2.6 File/Table vs. the Oracle7 Table

FoxPro 2.6 File/Table	Oracle7 Table
A table is a logical object that corresponds to a .DBF file.	A table is a logical object.
A file/table is the basic unit of storage in FoxPro 2.6. All data is contained in files.	A table is the basic unit of storage in Oracle7. All data is contained in tables.
A program must open a file/table before accessing its data.	A program connects to the database as a specified Oracle7 user, and accesses tables by naming them in SQL statements. The user must have database-level permission to access the tables.
A table corresponds to one operating system file.	Oracle7 provides a many-to-many correspondence between Oracle7 tables and operating system level files.
Space within FoxPro 2.6 database files is managed completely by the operating system.	Space within Oracle7 database files is managed completely by Oracle7.
Each table may optionally have indexes. To support high-speed access to records, FoxPro 2.6 uses these indexes. The application program determines which index to use.	An Oracle7 table may optionally have a primary key constraint, and one or more unique key constraints. These constraints each consist of an ordered set of one or more columns and do not affect the physical arrangement of rows in a table. To support high-speed access to records and to ensure uniqueness of key values, Oracle7 maintains an index for each primary or unique key constraint. (See the "FoxPro 2.6 Key and Index vs. Oracle Key, Index, and Constraint" table below.)
FoxPro 2.6 has no clustering or hashing.	In addition to the default storage methods used for most tables, Oracle7 offers two physical storage options that can improve data access performance in some cases: clustered tables and hash clusters. <i>Clustered tables</i> store rows from two or more logically related tables together, to minimize disk accesses when the rows are accessed together. <i>Hash clusters</i> enable rapid lookup by unique key value. Refer to Oracle7 documentation for more information.

FoxPro 2.6 File/Table	Oracle7 Table
Maximum file size: OS and hardware dependent.	Maximum file size: Hardware-dependent, but usually 2G or more.
	Maximum row size: A table can have up to 255 columns. At most one column per table can be of type LONG or LONG RAW, which can accommodate 2 GB. Therefore the maximum row size is over 2 GB.
FoxPro 2.6 has optimal pre-imaging where initial version of records are saved until the write operation is complete.	Standard Oracle mode of operating.
<p>Depending on the command, FoxPro 2.6 locks a record or a table file before the command is executed. If the command is meant to affect only one record, then that record is locked. Otherwise, the entire table is locked. This reduces concurrency. For example:</p> <p>Delete record <n> locks the record <n>.</p> <p>Delete <scope beyond one> locks the entire table.</p>	Row-level locking is standard. By default, readers never block and writers block each other only when contending for the same row.

FoxPro 2.6 Key and Index vs. Oracle7 Key, Index, and Constraint

FoxPro 2.6 Key and Index	Oracle7 Key, Index, and Constraint
<p>In FoxPro 2.6, multiple indexes can be created on a table. There is no concept of primary key or unique constraint. You have to specify the name of the index you want to use with your table.</p>	<p>In Oracle7, a <i>key</i> is the ordered set of one or more columns over which a primary key, unique key, foreign key, or referenced key integrity constraint is defined.</p>
<p>You might or might not have an index on a FoxPro 2.6 table. It's strongly recommended that you use structural compound indexes on a table.</p>	<p>Each Oracle7 table may optionally have one <i>primary key constraint</i>, and one or more <i>unique key constraints</i>. It is strongly recommended, but not mandatory for each a table to have a primary key constraint.</p> <p>An index is created and maintained by Oracle for each primary or unique key constraint. The index is a physically separate database object from the table that it indexes. It may belong to the same tablespace as the table, or to a different tablespace. You may drop any Oracle index.</p> <p>Additional unique or non-unique indexes may be created on any Oracle7 table using the SQL statement CREATE [UNIQUE] INDEX. The index is a physically separate database object from the table that it indexes. It may belong to the same tablespace as the table, or to a different tablespace.</p>
<p>Each FoxPro 2.6 record has a unique record number.</p>	<p>Each Oracle7 row of a non-clustered table has a unique physical address that is accessible through the pseudocolumn <i>ROWID</i>. The ROWID of a row is not permanent; it changes when the row is deleted and reinserted (e.g., through the SQL statements DELETE and INSERT, or by using the export and import utilities).</p>
<p>Each FoxPro 2.6 record can be uniquely referenced through its record number.</p>	<p>Each Oracle7 row can be uniquely referenced through its primary key, or any unique key, if these keys exist. Each row of a non-clustered table can be uniquely referenced by its ROWID, until the row is deleted and re-inserted (or exported and re-imported), at which time the ROWID will change.</p>

FoxPro 2.6 Database vs. Oracle7 Database

FoxPro 2.6 Database	Oracle7 Database
There is no concept of database in FoxPro 2.6. It's just a collection of files.	An Oracle7 database consists of one or more control files, 2 or more on-line redo log files, and one or more data files.
Space within a FoxPro 2.6 database file is completely managed by the operating system. As data is added to a file, the file may grow.	Space within an Oracle7 database file is completely managed by Oracle7. Database data files are preallocated with a specified size. After a file is created, its size can only be changed using Oracle7 Version 7.2 or greater. Space can be added to a database by adding a new data file to a tablespace.
FoxPro 2.6 has no concept analogous to the Oracle7 tablespace.	Each table belongs to a particular tablespace , which consists of one or more files. Multiple tables can belong to the same tablespace. Therefore a table can be spread across one or more files, and a database file can contain portions of one or more tables. Data storage in Oracle7 is highly flexible and controllable, so that performance can be optimized and administrative activities can be better organized. The physical storage arrangement is completely transparent to the application programs.
FoxPro 2.6 does not have transaction control.	Oracle always uses rollback segments for transaction control. The entire process is performed by the database without programmer intervention.

APPENDIX

A

ORACLE7 NORMALIZED SCHEMA

This appendix covers the Oracle7 schema for the normalized tables used in the intermediate Oracle7 database.

Table: user_info

This table will hold the user information.

Column Name	Comments
user_name	
user_password	User password in the source database.
temp_password	Temporary Oracle Password, which will be used to generate the “connect” statements to create all the schema objects owned by this user.
real_password	Real Oracle Password for the user. One of the scripts will include “alter user” statements.
def_tblsp	Default tablespace.
temp_tblsp	Temporary tablespace.
profile_name	Name of the resource profile assigned to the user.

Table: obj_info

This table will hold the objects' information.

Column Name	Comments
obj_id	Unique identifier for an object.
obj_name	Name of the object.
obj_type	Type of object; 'TBL' for table, 'VIW' for view, 'SPR' for stored procedure, 'TRI' for trigger, 'IDX' for index, 'PRS' for private synonym, and 'PUS' for public synonym.
obj_owner	Owner of the object.

Table: tbl_info

This table will contain the table information.

Column Name	Comments
table_id	Unique identifier for the table.
tablesp_name	Database name from the source database which can be overwritten with the tablespace name.
cluster_name	
pct_free	
pct_used	
ini_trans	
max_trans	
initial_extent	
next_extent	
min_extent	
max_extent	
pct_increase	

Table: tbl_col_info

This table will contain the table-column information.

Column Name	Comments
table_id	Unique identifier for the table.
col_name	Column name.
changed_col_name	We may need to change the original column name as it may be one of the Oracle7 reserved words.
user_datatype	Name of the user-defined datatype (if any)
src_datatype	This datatype is from the source database. This should be the base datatype when the source datatype is user-defined.
src_length	Length of the column in source database. In case of decimal datatype it represents precision.
src_scale	Scale of the column in case of decimal number.
ora_length	This will default to src_length.
ora_scale	This will default to src_scale.
src_order	Order of the column.
ora_order	This will default to src_order.
null_flag	Indicates whether column allows null values or not.
src_defnm	Default attached to the column, if any.

Table: datatype_info

This table will hold the datatype conversion information.

Column Name	Comments
src_datatype	Source database datatype.
ora_datatype	Oracle7 database datatype.
length_flag	0 - do not see ORA_LENGTH or ORA_SCALE. 1 - See ORA_LENGTH only. 2 - See both ORA_LENGTH and ORA_SCALE.

Table: ora_reserved_words

This table will hold the Oracle7 reserved words.

Column Name	Comments
reserved_word	A reserved word.

Table: command_generator

This table will hold the Oracle7 DDLs.

Column Name	Comments
command	A line of DDL.
seq	The order in which DDL's will be executed.

Table: control_generator

This table will hold the control file script.

Column Name	Comments
command	A line of control file script.
seq	The order in which lines are written to control file for a table.
tab	Identifies the tables within which control lines are to be grouped.

Table: list_generator

This table will hold the commands to generate control files.

Column Name	Comments
command	Commands which will generate control files and spool control statements in the files.
seq	The order in which commands will be executed.

Table: statement_generator

This table will hold the SQLLOAD commands.

Column Name	Comments
command	SQLLOAD commands.
seq	The order in which commands will be executed.

APPENDIX

B

SYSTEM TABLES

This appendix lists and describes all of the Oracle7 system tables.

FoxPro 2.6

FoxPro 2.6 does not have system catalogs. A file <filename.DBF> exists for each and every structure or table. That file contains data and information about columns.

ORACLE7

Oracle7 stores the data dictionary information in system catalogs. Various views are created on these system catalogs during database creation. These views greatly simplify access to the data dictionary.

There are three types of views. The views for a particular user which give information about the objects created by that user. The views that give the information about the schema objects accessible to a particular user. And the views used by the DBA, which give the information about all the schema objects present in the database.

1. The views that extract the information about the objects owned by a particular user:

Name of the View	Comments
USER_AUDIT_OBJECT	Audit trail records for statements concerning objects.
USER_AUDIT_SESSION	All audit trail records concerning connections and disconnections for the user.
USER_AUDIT_STATEMENT	Audit trail entries for the following statements issue by the user: GRANT, REVOKE, AUDIT, NOAUDIT, ALTER SYSTEM.
USER_AUDIT_TRAIL	Audit Trail entries relevant to the user.
USER_CATALOG	Tables, views, synonyms and sequences owned by the user.
USER_CLUSTERS	Description of user's own clusters.
USER_CLU_COLUMNS	Mapping of columns in user's tables to cluster columns.
USER_COL_COMMENTS	Comments on columns of user's tables and views.

Name of the View	Comments
USER_COL_PRIVS	Grants on columns for which the user is the owner, grantor, or the grantee.
USER_COL_PRIVS_MADE	All grants on columns of objects owned by the user.
USER_COL_PRIVS_RECD	Grants on columns for which the user is the grantee.
USER_CONSTRAINTS	Constraint definitions on user's tables.
USER_CONS_COLUMNS	Information about columns in constraint definitions owned by the user.
USER_DB_LINKS	Database links owned by the user.
USER_DEPENDENCIES	Dependencies to and from a user's objects.
USER_ERRORS	Current errors on all user's stored objects.
USER_EXTENTS	Extents of the segments belonging to a user's objects.
USER_FREE_SPACE	Free extents in tablespaces accessible to the user.
USER_INDEXES	Description of the user's own indexes.
USER_IND_COLUMNS	Columns of the user's indexes on user's tables.
USER_OBJECTS	Objects owned by the user
USER_OBJECT_SIZE	Size of the code of user's PL/SQL objects.
USER_OBJ_AUDIT_OPTS	Auditing options for user's own tables and views.
USER_RESOURCE_LIMITS	Displays the resource limits for the current user.
USER_ROLE_PRIVS	Roles granted to the user.
USER_SEGMENTS	Storage allocation for database segments belonging to a user's objects.
USER_SEQUENCES	Description of the user's own sequences.
USER_SNAPSHOTS	Snapshots the user can view.

Name of the View	Comments
USER_SNAPSHOT_LOGS	All snapshot logs owned by the user.
USER_SOURCE	Text source of all stored objects belonging to the user.
USER_SYNONYMS	The user's private synonyms.
USER_SYS_PRIVS	System privileges granted to the user.
USER_TABLES	Description of the user's own tables.
USER_TABLESPACES	Description of accessible tablespaces.
USER_TAB_COLUMNS	Columns of user's tables, views and clusters.
USER_TAB_COMMENTS	Comments on the tables and views owned by the user.
USER_TAB_PRIVS	Grants on objects for which the user is the owner, grantor, or grantee.
USER_TAB_PRIVS_MADE	All grants on objects owned by the user.
USER_TAB_PRIVS_RECD	Grants on objects for which the user is the grantee.
USER_TRIGGERS	Description of user's own triggers.
USER_TRIGGER_COLS	Shows usage of columns in triggers owned by user or on one of user's tables.
USER_TS_QUOTA	Tablespace quotas for the user.
USER_USERS	Information about the current user.
USER_VIEWS	Text of views owned by the user.

2. The views that extract the information about the objects accessible through grants, to a particular user:

Name of the View	Comments
-------------------------	-----------------

Name of the View	Comments
ALL_TABLES	All tables, views, synonyms, and sequences accessible to the user.
ALL_COL_COMMENTS	Comments on columns of accessible tables and views.
ALL_COL_PRIVS	Grants on columns for which the user or PUBLIC is the grantee.
ALL_COL_PRIVS_MADE	Grants on columns for which the user is owner or grantor.
ALL_TAB_PRIVS_RECD	Grants on objects for which the user or PUBLIC is the grantee.
ALL_CONSTRAINTS	Constraint definitions on accessible tables.
ALL_CONS_COLUMNS	Information about accessible columns in constraint definitions.
ALL_DB_LINKS	Database links accessible to the user.
ALL_DEF_AUDIT_OPTS	Default object-auditing options that will be applied when objects are created.
ALL_DEPENDENCIES	Dependencies between objects, accessible to the user.
ALL_ERRORS	Current errors on all objects accessible to the user.
ALL_INDEXES	Description of indexes accessible to the user.
ALL_IND_COLUMNS	Columns of the indexes accessible to the user.
ALL_OBJECTS	Objects accessible to the user
ALL_SEQUENCES	Description of sequences accessible to the user.
ALL_SNAPSHOTS	All snapshots accessible to the user.
ALL_SOURCE	Text source of all stored objects accessible to the user.
ALL_SYNONYMS	All synonyms accessible to the user.

Name of the View	Comments
ALL_TABLES	Description of tables accessible to the user.
ALL_TAB_COLUMNS	Columns of all tables, views and clusters accessible to the user.
ALL_TAB_COMMENTS	Comments on the tables and views accessible to the user.
ALL_TAB_PRIVS	Grants on objects for which the user or PUBLIC is the grantor, or grantee.
ALL_TAB_PRIVS_MADE	User's grants and grants on user's objects.
ALL_TAB_PRIVS_RECD	Grants on objects for which the user or PUBLIC is the grantee.
ALL_TRIGGERS	Triggers accessible to the user.
ALL_TRIGGER_COLS	Shows usage of columns in triggers owned by the user, on tables owned by the user.
ALL_USERS	Information about all users of the database.
ALL_VIEWS	Text of views accessible to the user.

3. The views that extract the information about the objects accessible to DBA. These views can be executed only by DBA:

Name of the View	Comments
DBA_2PC_NEIGHBORS	Information about incoming and outgoing connections for pending transactions.
DBA_2PC_PENDING	Information about failed distributed transactions in PREPARED state.
DBA_AUDIT_EXISTS	Audit trail entries created by the AUDIT EXISTS command.
DBA_AUDIT_OBJECT	All audit trail records for all objects in the system.

Name of the View	Comments
DBA_AUDIT_SESSION	All audit trail records in the system concerning CONNECT and DISCONNECT.
DBA_AUDIT_STATEMENT	All audit trail records in the system concerning GRANT, REVOKE, AUDIT, NOAUDIT, and ALTER SYSTEM statements.
DBA_AUDIT_TRAIL	Collection of all audit trails in the system.
DBA_BLOCKERS	All sessions that have someone waiting on a lock they hold that are not themselves waiting on a lock.
DBA_CATALOG	All database tables, views, synonyms, and sequences.
DBA_CLUSTERS	Description of all clusters in the database.
DBA_CLU_COLUMNS	Mapping of table columns to cluster columns.
DBA_COL_COMMENTS	Comments on columns of all tables and views.
DBA_COL_PRIVS	All grants on columns in the database.
DBA_CONSTRAINTS	Constraint definitions on all tables in the database.
DBA_CONS_COLUMNS	Information about all columns in constraint definitions.
DBA_DATA_FILES	Information about database files.
DBA_DB_LINKS	All database links in the database.
DBA_DDL_LOCKS	All DDL locks held in the database and all outstanding requests for a DML lock.
DBA_DEPENDENCIES	Dependencies to and from all objects.
DBA_DML_LOCKS	All DML locks held in the database and all outstanding requests for a DML lock.
DBA_ERRORS	Current errors on all stored objects in the database.
DBA_EXP_FILES	Description of export files.

Name of the View	Comments
DBA_EXP_OBJECTS	Objects that have been incrementally exported.
DBA_EXP_VERSIONS	Version number of the last export session.
DBA_EXTENTS	Extents of all segments in the database.
DBA_FREE_SPACE	Free extents in all tablespaces.
DBA_INDEXES	Description of all indexes in the database.
DBA_IND_COLUMNS	Columns of the indexes on all tables and clusters.
DBA_LOCKS	All locks or latches held in the database, and all outstanding requests for a lock or latch. This view includes DML locks and DDL locks.
DBA_OBJECTS	All clusters, database links, indexes, packages, package bodies, sequences, synonyms, tables, and views defined in the database.
DBA_OBJECT_SIZE	Size of the code of all PL/SQL objects in the database.
DBA_OBJ_AUDIT_OPTS	Auditing options for all tables and views.
DBA_PRIV_AUDIT_OPTS	Privilege auditing options. (One row for each audited privilege)
DBA_PROFILES	Resource limits assigned to each profile.
DBA_ROLES	All roles that exist in the database.
DBA_ROLES_PRIVS	Description of roles granted to users and to roles.
DBA_ROLLBACK_SEGS	Description of rollback segments.
DBA_SEGMENTS	Storage allocated for all database segments.
DBA_SEQUENCES	Description of all sequences in the database.
DBA_SNAPSHOTS	All snapshots in the database.
DBA_SNAPSHOT_LOGS	All snapshots logs in the database.

Name of the View	Comments
DBA_SOURCE	Text source of all stored objects in the database.
DBA_STMT_AUDIT_OPTS	Describes current system auditing options across the system and by user.
DBA_SYNONYMS	All synonyms in the database.
DBA_SYS_PRIVS	Description of system privileges granted to the users and to roles.
DBA_TABLES	Description of all tables in the database.
DBA_TABLESPACES	Description of all tablespaces in the database.
DBA_TAB_COLUMNS	Columns of all tables, views and clusters.
DBA_TAB_COMMENTS	Comments on all the tables and views in the database.
DBA_TAB_PRIVS	All grants on objects in the database.
DBA_TRIGGERS	Description of all the triggers in the database.
DBA_TRIGGER_COLS	Shows usage of columns in triggers defined by any user, on any user's table .
DBA_TS_QUOTAS	Tablespace quotas for all users.
DBA_USERS	Information about all users of the database.
DBA_VIEWS	Text of all views in the database.
DBA_WAITERS	All sessions waiting for locks and the session that holds the lock.

In addition to these views Oracle7 databases externalize a large amount of dynamic information. This information is accessible through a set of views starting with V\$ and is primarily used for performance tuning. These views are continuously updated while a database is open and in use.

Recommendations

FoxPro 2.6 does not support system catalogs, which is an essential entity for any RDBMS. Oracle7 stores data dictionary information in System Catalogs. It's

very easy to query all these system catalogs in Oracle7 using simple SQL. So, Oracle7 exceeds the functionality of FoxPro 2.6 in keeping data dictionary information.

APPENDIX

C

CONVERSION ESTIMATION GUIDE

The following guide can help you estimate the time and processes involved in your FoxPro 2.6 to Oracle conversion. We would like to hear from you about the enclosed estimates and any other comments you think would be helpful to us for future upgrades to this document.

Oracle from FoxPro 2.6 Conversion Project Conversion Estimation Guide

Please tell us about your company and the product to be converted:

Company Name: _____

Your Name: _____

Date: _____

Name of product to be converted: _____

Summary of product functionality:

Please tell us about your database:

What version of FoxPro are you currently using? _____

How many of each of the following database objects exist in your application?

Tables: _____

Tables with a MEMO column: _____

Tables with two or more MEMO columns: _____

Tables with a GENERAL column: _____

Tables with two or more GENERAL columns: _____

What is the total size of your development database (in MB): _____

Indexes: _____

Number of Indexes: _____

Please tell us about your application:

The following table will help us understand your application. For each tool that you use to access the database, please fill in one line in the table.

Tool name	Version	Percent of Application built with this tool.	Number of functions/ programs	Number of reports	Number of screens	Length of code for each function, report or screen
Example: Visual BASIC	2.0	80%			20	300 Lines
Visual BASIC						
Visual C++						
FoxPro 2.6						
Other						

Please include example code for each of the tools listed above.

Is there any additional information that you would like to add about the tools used to create your application?

Please help us estimate the amount of effort required for your conversion:

Conversion Step	Quantity of objects to be converted	Quantity per day multiplier*	Estimated number of days*
Tables			3

DATABASE SCHEMA TOTAL

TOTAL			

DATABASE SCHEMA TOTAL	
TOTAL	
TESTING	
TOTAL	

