

Informix Migration Guide

for Database Servers

Version 9.1
March 1997
Part No. 000-4800

Published by INFORMIX® Press

Informix Software, Inc.
4100 Bohannon Drive
Menlo Park, CA 94025

Copyright © 1981-1997 by Informix Software, Inc. or their subsidiaries, provided that portions may be copyrighted by third parties, as set forth in documentation. All rights reserved.

The following are worldwide trademarks of Informix Software, Inc., or its subsidiaries, registered in the United States of America as indicated by “®,” and in numerous other countries worldwide:

INFORMIX®; C-ISAM®; INFORMIX®-OnLine Dynamic Server™; DataBlade®

The following are worldwide trademarks of the indicated owners or their subsidiaries, registered in the United States of America as indicated by “®,” and in numerous other countries worldwide:

Adobe Systems Incorporated: PostScript®
Information Builders, Inc.: EDA/SQL™;
International Business Machines Corporation: IBM®; DB2®; IMS®; MVS™; DRDA™
Microsoft Corporation: Microsoft®; MS®; MS-DOS®; CodeView®; MS Windows™; Windows™;
Windows NT™; Windows® 95; ODBC™; Visual Basic™; Visual C++™
Oracle Corporation: ORACLE®; ORACLE7 Server™
Sybase, Inc.: SYBASE®; SYBASE SQL Server 10™
X/Open Company Ltd.: UNIX®; X/Open®

All other marks or symbols are registered trademarks or trademarks of their respective owners.

Documentation Team: Twila Booth, Evelyn Eldridge-Diaz, Karen Smith

To the extent that this software allows the user to store, display, and otherwise manipulate various forms of data, including, without limitation, multimedia content such as photographs, movies, music and other binary large objects (blobs), use of any single blob may potentially infringe upon numerous different third-party intellectual and/or proprietary rights. It is the user's responsibility to avoid infringements of any such third-party rights.

RESTRICTED RIGHTS/SPECIAL LICENSE RIGHTS

Software and documentation acquired with US Government funds are provided with rights as follows: (1) if for civilian agency use, with Restricted Rights as defined in FAR 52.227-19; (2) if for Dept. of Defense use, with rights as restricted by vendor's standard license, unless superseded by negotiated vendor license as prescribed in DFAR 227.7202. Any whole or partial reproduction of software or documentation marked with this legend must reproduce the legend.

Table of Contents

Introduction

About This Manual	3
Organization of This Manual	4
Types of Users	5
Software Dependencies	6
Assumptions About Your Locale.	6
Demonstration Database	7
Documentation Conventions	7
Typographical Conventions	8
Icon Conventions	9
Command-Line Conventions	11
Sample-Code Conventions.	14
Additional Documentation	14
On-Line Manuals	15
Printed Manuals	15
On-Line Help	15
Error Message Files	16
Documentation Notes, Release Notes, Machine Notes	16
Compliance with Industry Standards	17
Informix Welcomes Your Comments	18

Section I Migrating Data

Chapter 1 Data Migration Overview

Planning for Data Migration	1-3
Migrating Between Versions	1-3
Moving to a Different Database Server or Environment	1-4
Migrating Client Applications	1-4
Importing Data from Other Sources.	1-4
Overview of Informix Database Servers	1-5

Where to Find Information on Migrating to a Database Server . . .	1-6
Upgrading Database Servers.	1-6
Reverting to an Earlier Version	1-7
Migrating Between Database Servers in the Same Environment	1-7
Migrating to a Different Database Server Environment.	1-8
Tools for Moving Data	1-9
Choosing the Appropriate Migration Utility	1-11

Chapter 2 Moving Between Computers

Moving Database Server Data Between Computers Using onunload and onload	2-3
Steps for Using onunload and onload.	2-4
Moving Data to a New Environment	2-7
Importing Data from a Non-Informix Environment	2-8
Using the Informix Gateway Products	2-8
Using the High-Performance Loader	2-9

Section II Upgrading Versions of Database Servers

Chapter 3 Upgrading OnLine Dynamic Server 7.2 to Universal Server

Preparing to Migrate to Universal Server	3-3
Changes Introduced by Universal Server	3-4
Environment Variables.	3-4
Configuration Parameters.	3-5
New Database Storage Spaces	3-6
Global Language Support.	3-6
Migrating from OnLine Dynamic Server 7.2 to Universal Server . . .	3-6
Check Available Space	3-8
Save Copies of the Current Configuration Files	3-9
Close All Transactions	3-10
Initiate a Fast Recovery.	3-10
Verify the Integrity of the Data	3-11
Verify the Mode	3-12
Make a Final Backup	3-12
Bring OnLine Dynamic Server Off-Line	3-13
Reconfigure the UNIX Operating System	3-14
Install Universal Server	3-14
Verify That Environment Variables Are Set Correctly	3-15
Update the ONCONFIG Configuration Files	3-15
Add a Communications Support Module (Optional)	3-16

Update the ON-Archive Configuration Files	3-16
Bring Universal Server On-Line	3-17
Verify the Integrity of the Data	3-18
Make an Initial Universal Server Backup	3-18
Migration Complete	3-19
Install and Configure DataBlade Modules.	3-19
Reverting Universal Server to OnLine Dynamic Server 7.2	3-20
Determine Whether Reversion is Possible.	3-21
Save Copies of the Configuration Files	3-21
Verify the Integrity of the Data	3-22
Back Up the Universal Server Data	3-22
Remove Universal Server Features	3-22
Run the Reversion Utility	3-26
Modify Configuration Parameters	3-27
Reset Environment Variables	3-27
Remove Communications Support Module Settings	3-28
Install and Start OnLine Dynamic Server	3-28
Verify the Integrity of the Data	3-28
Back Up OnLine Dynamic Server Data.	3-28
Return OnLine Dynamic Server to On-Line Mode	3-28

Chapter 4 **Migrating OnLine Dynamic Server 6.0 or Later to 7.2x**

Changes Introduced by OnLine Dynamic Server	4-6
Environment Variable Changes in Version 7.2x	4-6
Configuration Parameter Changes in Version 7.2x	4-8
Global Language Support Changes in Version 7.2x	4-9
Environment Variable Changes in Version 7.1UD1.	4-9
Configuration Parameter Changes in Version 7.1UD1.	4-10
Changes to BlobSpace Requirements in Version 7.1UD1	4-11
Environment Variable Changes in Version 7.1	4-12
Configuration Parameter Changes in Version 7.1	4-12
ON-Archive Changes in Version 7.1	4-12
Environment Variable Changes in Version 6.0	4-13
Preparing to Migrate Between Versions	4-13
Install the Latest Maintenance Release for the Current Version	4-14
Check Available Space and System Requirements	4-15
Save Copies of the Current Configuration Files	4-15
Close all Transactions.	4-16
Initiate a Fast Recovery	4-17
Verify the Integrity of the Data	4-17
Back Up Your OnLine Dynamic Server Files	4-18
Preparation Complete	4-19

Upgrading to OnLine Dynamic Server Version 7.2x: UNIX	4-19
Bring OnLine Dynamic Server Off-Line	4-20
Reconfigure the UNIX Operating System	4-20
Install OnLine Dynamic Server 7.2x (UNIX)	4-20
Verify That Environment Variables Are Set Correctly	4-21
Update the ONCONFIG Configuration Files	4-21
Update the ON-Archive Configuration Files	4-22
Configure the Database Server to Use Enterprise Replication (Version 7.22 Only)	4-22
Bring OnLine Dynamic Server 7.2x On-Line	4-23
Verify the Integrity of the Data	4-24
Make an Initial OnLine Dynamic Server 7.2x Backup	4-24
Run UPDATE STATISTICS	4-24
Verify the Access Path of Your SQL Statements	4-25
Migration Complete.	4-25
Upgrading OnLine Dynamic Server to Version 7.22:	
Windows NT	4-26
Bring OnLine Dynamic Server Off-Line	4-26
Install the Database Server and the Administration Tools	4-27
Customize the Database Server Environment	4-28
Configure the Database Server to Use Workgroup Replication (Version 7.22 Only)	4-29
Bring OnLine Dynamic Server On-Line	4-29
Verify the Integrity of the Data	4-30
Make a Complete Backup of OnLine Dynamic Server	4-30
Run Update Statistics	4-30
Verify the Access Path of Your SQL Statements	4-31
Migration Complete.	4-31
Changing Database Server Definitions	4-32
Migrating to a GLS Locale	4-32
Reverting to OnLine Dynamic Server: UNIX	4-33
Remove Unsupported SQL Features	4-34
Remove Fragmented Tables (Version 6.0 Only)	4-34
Save Catalog Information	4-34
Stop Enterprise Replication (Version 7.22 Only)	4-35
Run the Reversion Utility	4-35
Remove GLS Features	4-36
Modify Configuration Parameters	4-37
Reset Environment Variables.	4-40
Modify the sqlhosts File	4-41
Rename ON-Archive Files (Version 6.0 Only)	4-42

Start the Desired Version of OnLine Dynamic Server	4-42
Restore Catalog Information (Version 6.0 Only)	4-42
Verify the Integrity of the Data	4-42
Back Up OnLine Dynamic Server Data	4-43
Return OnLine Dynamic Server to On-Line Mode	4-43
Reverting from OnLine Dynamic Server 7.1x to 7.1	4-43
Reverting OnLine Dynamic Server: Windows NT	4-44
Remove Unsupported SQL Features	4-44
Uninstall Workgroup Replication Manager (Version 7.22 Only)	4-45
Run the Reversion Utility	4-45
Uninstall OnLine Dynamic Server and Administration Tools . .	4-46
Remove GLS Features	4-46
Modify Configuration Parameters	4-47
Reset Environment Variables	4-47
Reinstall the Earlier Version of OnLine Dynamic Server	4-48
Bring OnLine Dynamic Server On-Line	4-49
Verify the Integrity of the Data	4-50
Back Up the OnLine Dynamic Server Data	4-50
Reversion is Now Complete	4-50

Chapter 5

Migrating Pre-6.0 to 7.2x OnLine Dynamic Server: UNIX

Preparation for Migration Between Versions	5-3
Changes Introduced in Version 6.0	5-4
Changes to Database Utilities	5-5
Planning for Migration	5-6
Operating-System Configuration Issues	5-6
Estimating the Size and Number of Shared-Memory Segments	5-6
Configuring Semaphore Parameters	5-10
Configuring the Number of Open-File Descriptors	5-12
Operating-System Updates	5-12
Disk-Utilization Issues	5-13
Accommodating System Catalog Indexes	5-14
Accommodating the sysmaster Database	5-15
Locating Temporary Files and Tables	5-15
Accommodating Data Distributions	5-16
Accommodating User-Table Indexes	5-17
Accommodating the Conversion of User-Table Indexes	5-18
Managing Secure-Auditing Log Files	5-20
Database System-Configuration Issues	5-21
Changing Environment Variables	5-21
Revising Configuration Parameter Values	5-21

Increasing Logical-Log Space	5-22
Planning for Additional Shared-Memory Usage	5-22
Saving Pre-Existing sysmaster Databases	5-23
Configuring Secondary Database Servers for Data Replication	5-23
Managing Backups	5-23
Making Performance Comparisons	5-23
Migration from OnLine 4.1 or 5.0 on UNIX to OnLine Dynamic Server 6.0 or Later	5-24
Install the Latest Maintenance Release for the Current Version	5-24
Capture Configuration and Chunk-Layout Information	5-25
Close All Transactions and Make a Final Level-0 Backup	5-26
Bring OnLine 4.1 or 5.0 Off-Line	5-28
Install and Configure OnLine Dynamic Server 7.2x	5-29
Reconfigure the UNIX Operating System	5-31
Bring OnLine Dynamic Server 7.2x On-Line	5-32
Convert User-Table Indexes	5-33
Verify the Integrity of the Data	5-37
Migration Complete.	5-38
Global Language Support.	5-38
Reverting to OnLine 5.0	5-39
Save Configuration Files	5-39
Remove All Users from OnLine Dynamic Server	5-39
Verify the Integrity of the Data	5-40
Back Up OnLine Dynamic Server Data	5-40
Remove Features Introduced by Later Versions of OnLine Dynamic Server	5-40
Execute the Reversion Utility	5-41
Prepare the TBCONFIG Configuration File	5-42
Reset Environment Variables.	5-42
Bring Up the Desired Version of OnLine Server	5-42
Verify the Integrity of the Data	5-43
Back Up OnLine 5.0 Server Data	5-43
Return OnLine 5.0 to On-Line Mode	5-43

Chapter 6 Migrating OnLine Workgroup Server

How to Use This Chapter	6-3
OnLine Workgroup Server Differences	6-3
Using Different Server and Client Configurations	6-4
OnLine Workgroup Server 7.22 and Command Center 7.22	6-4
OnLine Workgroup Server 7.12 and Command Center 7.22	6-4
OnLine Workgroup Server 7.22 and Command Center 7.12	6-4
OnLine Workgroup Server 7.12 and Command Center 7.12	6-4

Preparing to Migrate Between Versions	6-5
Check Available Space and System Requirements	6-6
Save a Copy of the Current Configuration Files	6-6
Place the Database Server in Administration Mode	6-7
Verify the Integrity of the Data	6-8
Back Up Your OnLine Workgroup Server Data	6-9
Preparation Complete	6-9
Upgrading OnLine Workgroup Server to Version 7.2x	6-9
Bring OnLine Workgroup Server Off-Line	6-10
Install the Database Server and Administration Tools (Windows NT)	6-10
Install the Database Server (UNIX)	6-11
Customize the Database Server Environment	6-13
Configure the Database Server to Use Workgroup Replication.	6-13
Bring OnLine Workgroup Server On-Line.	6-14
Verify the Integrity of the Data	6-14
Make a Complete Backup of OnLine Workgroup Server.	6-15
Run Update Statistics.	6-15
Verify the Access Path of Your SQL Statements	6-15
Migration Complete	6-16
Migrating to a GLS Locale	6-16
Changing Database Server Definitions	6-16
Reverting OnLine Workgroup Server and Administration Tools	6-17
Remove Unsupported SQL Features	6-17
Uninstall Workgroup Replication Manager (Version 7.22 Only)	6-18
Run the Reversion Utility	6-18
Uninstall OnLine Workgroup Server and Administration Tools (Windows NT)	6-19
Remove GLS Features	6-19
Modify Configuration Parameters	6-20
Reset Environment Variables	6-21
Reinstall the Earlier Version of OnLine Workgroup Server (Windows NT)	6-22
Reinstall the Earlier Version of OnLine Workgroup Server (UNIX)	6-22
Bring OnLine Workgroup Server On-Line.	6-23
Verify the Integrity of the Data	6-23
Back Up the OnLine Workgroup Server Data	6-23
Reversion is Now Complete	6-23

Chapter 7	Migrating SE	
	Importing and Exporting Data	7-3
	Migrating Between Different Versions of SE	7-4
	Preparing to Migrate from a Pre-6.0 SE to SE 7.2x	7-4
	Preparing to Migrate from a Post-6.0 SE to SE 7.2x	7-5
	Reverting to an Earlier Version of SE	7-6
	Converting C-ISAM Files to SE	7-6
	Migrating from C-ISAM to OnLine Dynamic Server or OnLine Workgroup Server	7-7
 Chapter 8	 Moving Between Database Servers	
	Choosing a Migration Method	8-5
	Moving Data from SE to OnLine Dynamic Server or OnLine Workgroup Server	8-6
	Using the dbexport and dbimport Utilities	8-6
	Using the UNLOAD Statement	8-8
	Adapting Your Programs for OnLine Dynamic Server or OnLine Workgroup Server	8-11
	Moving Data from OnLine Dynamic Server or OnLine Workgroup Server to SE	8-12
	Eliminating Features That SE Does Not Support	8-12
	Using the dbexport and dbimport Utilities	8-12
	Using the UNLOAD Statement	8-14
	Adapting Your Programs for SE.	8-16
	Moving Data from OnLine Dynamic Server to OnLine Workgroup Server	8-17
	Moving Data in the Same Environment	8-17
	Moving Data Between Different Environments	8-17
	Save a Copy of the Current Configuration Files	8-18
	Verify the Integrity of the Data	8-18
	Back Up OnLine Dynamic Server	8-19
	Export the OnLine Dynamic Server Databases.	8-19
	Bring OnLine Dynamic Server Off-Line	8-19
	Install and Configure OnLine Workgroup Server.	8-20
	Verify Port Numbers and Services File	8-20
	Customize the Database Server Environment	8-20
	Bring OnLine Workgroup Server On-Line	8-21
	Import the Databases to OnLine Workgroup Server.	8-21
	Verify the Integrity of the Data	8-22
	Back Up OnLine Workgroup Server	8-22
	Run Update Statistics	8-22
	Migration Complete.	8-22
	Adapting Your Programs for OnLine Workgroup Server	8-23

Using the dbexport and dbimport Utilities	8-23
Using the UNLOAD Statement	8-24
Moving Data from OnLine Workgroup Server to	
OnLine Dynamic Server	8-26
Moving Data in the Same Environment	8-26
Moving Data Between Different Environments	8-27
Save a Copy of the Current Configuration Files	8-28
Verify the Integrity of the Data	8-28
Back Up OnLine Workgroup Server	8-28
Export the OnLine Workgroup Server Databases	8-29
Shut Down OnLine Workgroup Server (UNIX)	8-29
Shut Down OnLine Workgroup Server (Windows NT)	8-29
Install and Configure OnLine Dynamic Server and the	
Administration Tools (Windows NT)	8-29
Install OnLine Dynamic Server (UNIX)	8-30
Verify Port Numbers and Services File	8-30
Customize the Database Server Environment	8-30
Bring OnLine Dynamic Server On-Line	8-31
Import the Databases into OnLine Dynamic Server	8-31
Verify the Integrity of the Data	8-31
Back Up OnLine Dynamic Server	8-32
Run Update Statistics	8-32
Migration Complete	8-32
Adapting Your Programs for OnLine Dynamic Server	8-32
Using the dbexport and dbimport Utilities	8-33
Using the UNLOAD Statement	8-34
Moving OnLine Dynamic Server Data Between Environments	8-36
Procedure for Moving Data Between Environments	8-36
Using the Migration Utilities	8-37
Adapting Your Programs for the UNIX or NT Environment	8-37
Migration Complete	8-37
Moving OnLine Workgroup Server Data Between Environments	8-38
Procedure for Moving Data Between Environments	8-38
Using the Migration Utilities	8-39
Adapting Your Programs for the UNIX or NT Environment	8-39
Migration Complete	8-39

Chapter 9

Changing Locales

Understanding Database-to-Database Server Compatibility	9-3
Selecting the Database Locale	9-4
Finding the Locale of an Asian (ALS) Database	9-6
Finding the Locale of a Pre-6.0 Database	9-6
Finding the Locale of a 6.x, 7.1, or 7.1x Database	9-6

Finding the Available Locales for Version 7.2x or Universal Server Databases	9-7
Environment Variables Used for Language Support	9-8
Native Language Support (NLS)	9-8
Asian Language Support (ALS)	9-10
Global Language Support (GLS)	9-12
Changing from an NLS Locale to a GLS Locale	9-14
Migrating from a Version 6.x or 7.1x Product to Version 7.2x	9-14
Changing from an ALS Locale to a GLS Locale	9-17
Migrating from Version 6.x ALS Products to Version 7.2x	9-17
Migrating from Version 5.x ALS Products to Version 7.2x	9-18
Migrating from Version 4.x ALS Products to Version 7.2x	9-19
Migrating from Version 4.x ASCII Japanese Products to Version 7.2x	9-20
Reverting from a GLS Locale to an NLS Locale	9-20
Reverting to OnLine Dynamic Server Version 6.0, 7.1x, or OnLine Workgroup Server 7.12.	9-21
Reverting from SE 7.2x to an Earlier Version	9-21
Reverting from a GLS Locale to an ALS Locale	9-22
Reverting from Version 7.2x to Version 6.x ALS	9-22
Reverting from Version 7.2x to Version 5.x ALS	9-23
Reverting from Version 7.2x to Version 4.x ALS or ASCII Product	9-24

Section III Reference

Chapter 10 Utilities for Data Migration

Setting Environment Variables	10-4
The dbexport Utility	10-5
Using the -c Option	10-7
Using the -ss Option.	10-7
Destination Options.	10-8
The Contents of the Schema File	10-10
Exporting Simple Large Objects.	10-10
The dbimport Utility	10-12
Using the -c Option	10-14
Using Delimited Identifiers with dbimport	10-14
Canceling dbimport	10-14
Database-Logging Mode	10-15
Input-File Location Options	10-15
Create Options	10-17

Using the -l Options	10-18
Changing the Database Name	10-19
Importing Simple Large Objects	10-20
Using dbimport to Change the Locale of a Database	10-20
The dbload Utility	10-21
Using the dbload Utility	10-23
Choosing Between dbload and LOAD	10-25
Creating a dbload Command File	10-25
The dbschema Utility	10-37
Creating the Schema for a Database	10-39
Obtaining the Role Schema	10-46
Displaying the Distribution Information for Tables	10-49
The onload Utility	10-54
Specify Source Parameters	10-55
Create Options	10-56
Constraints That Affect onload and onunload	10-57
Constraints That Affect onload	10-58
Logging While Using onload	10-58
Moving Simple Large Objects to a Blob space	10-59
Ownership and Privileges	10-59
Exclusive Locking During Load Operation	10-59
The onmode Utility	10-60
Preparing to Use the -b Option	10-60
Reverting to OnLine Version 5.0	10-61
Reverting to OnLine Dynamic Server Version 6.0	10-62
Reverting to Version 7.2	10-63
The onunload Utility	10-64
Syntax	10-64
Specify Destination Parameters	10-65
Constraints That Affect onunload	10-65
Unloading a Database or Table	10-66
Logging Mode	10-67
Locking During Unload Operation	10-67
The LOAD Statement	10-68
The UNLOAD Statement	10-69
onload: The High-Performance Loader Utility	10-70

Appendix A Database Server Environment Variables

Index

Introduction

About This Manual	3
Organization of This Manual	4
Types of Users	5
Software Dependencies	6
Assumptions About Your Locale	6
Demonstration Database	7
Documentation Conventions	7
Typographical Conventions	8
Icon Conventions	9
Comment Icons	9
Feature, Product, and Platform Icons	9
Compliance Icons	10
Command-Line Conventions	11
How to Read a Command-Line Diagram	13
Sample-Code Conventions	14
Additional Documentation	14
On-Line Manuals	15
Printed Manuals	15
On-Line Help	15
Error Message Files	16
Documentation Notes, Release Notes, Machine Notes	16
Compliance with Industry Standards	17
Informix Welcomes Your Comments	18

Read this introduction for an overview of the information provided in this manual and for an understanding of the documentation conventions used.

About This Manual

The *Informix Migration Guide* describes the procedures that you use when you move data from one location to another and when you migrate existing Informix databases to and from the following Informix database servers:

- INFORMIX-Universal Server
- INFORMIX-OnLine Dynamic Server
- INFORMIX-OnLine Workgroup Server
- INFORMIX-SE

This manual also describes how to use the data migration utilities.

Organization of This Manual

This manual includes the following chapters:

- This Introduction provides an overview of the manual and describes the documentation conventions used.
- Section I, “Migrating Data,” covers how to migrate between Informix database servers and different environments:
 - [Chapter 1, “Data Migration Overview,”](#) discusses the reasons that you might have for moving data, gives an overview of Informix database servers, and introduces the tools that you can use.
 - [Chapter 2, “Moving Between Computers,”](#) describes how to move between different physical equipment (computer and storage devices) and from non-Informix environments.
- Section II, “Upgrading Versions of Database Servers,” covers how to upgrade and revert Informix database servers, and migrating between different locales:
 - [Chapter 3, “Upgrading OnLine Dynamic Server 7.2 to Universal Server,”](#) describes the steps to upgrade to Universal Server from OnLine Dynamic Server, Version 7.2. This chapter also discusses how to revert from Universal Server to OnLine Dynamic Server.
 - [Chapter 4, “Migrating OnLine Dynamic Server 6.0 or Later to 7.2x,”](#) covers the steps to upgrade OnLine Dynamic Server, and the steps to revert to an earlier version of OnLine Dynamic Server.
 - [Chapter 5, “Migrating Pre-6.0 to 7.2x OnLine Dynamic Server: UNIX,”](#) covers the steps to upgrade to OnLine Dynamic Server 7.2x from pre-6.0 versions of OnLine Dynamic Server, and the steps to revert to an earlier version.
 - [Chapter 6, “Migrating OnLine Workgroup Server,”](#) describes the steps to upgrade to OnLine Workgroup Server, Version 7.22, and the steps to revert to an earlier version of OnLine Workgroup Server.

- ❑ [Chapter 7, “Migrating SE,”](#) covers how to migrate between versions of INFORMIX-SE.
- ❑ [Chapter 8, “Moving Between Database Servers,”](#) describes the procedures for how to migrate between Informix database servers and different operating systems.
- ❑ [Chapter 9, “Changing Locales,”](#) describes how to move databases between the Asian Language Support (ALS), Native Language Support (NLS), and Global Language Support (GLS) locales.
- Section III, “Reference,” contains reference material on the migration utilities and environment variables:
 - ❑ [Chapter 10, “Utilities for Data Migration,”](#) describes the syntax and use of utilities that you use to move data. The **dbexport**, **dbimport**, **dbload**, **dbschema**, **onmode**, and **onpload** utilities have been extended to support Universal Server databases.
 - ❑ [Appendix A](#) lists the environment variables that the Informix database servers use.

Types of Users

This manual is written for database administrators and system administrators who plan to upgrade an Informix database server or move data from one Informix database server to another.

This manual assumes that you are familiar with Informix database servers and the databases to and from which you will move data.

If you are not familiar with OnLine Dynamic Server or SE, read *Getting Started with Informix Database Server Products*. For information about Universal Server, read [Getting Started with INFORMIX-Universal Server](#).

Software Dependencies

This manual assumes that you are using one or more of the following database servers:

- INFORMIX-Universal Server
- INFORMIX-OnLine Dynamic Server
- INFORMIX-OnLine Workgroup Server
- INFORMIX-SE

In this manual, all instances of Universal Server refer to INFORMIX-Universal Server.

Assumptions About Your Locale

Informix products can support many languages, cultures, and code sets. All culture-specific information is brought together in a single environment, called a GLS (Global Language Support) locale.

This manual assumes that you are using the default locale, **en_us.8859-1**. This locale supports U.S. English format conventions for dates, times, and currency. In addition, this locale supports the ISO 8859-1 code set, which includes the ASCII code set plus many 8-bit characters such as é, è, and ñ.

If you plan to use nondefault characters in your data or your SQL identifiers, or if you want to conform to the nondefault collation rules of character data, you need to specify the appropriate nondefault locale(s). For instructions on how to specify a nondefault locale, additional syntax, and other considerations related to GLS locales, see the [Guide to GLS Functionality](#).

UNIX

NT

Demonstration Database

The DB-Access utility, which is provided with your Informix database server products, includes a demonstration database called **stores7** that contains information about a fictitious wholesale sporting-goods distributor. Sample command files are also included. The **stores7** database is described in detail and its contents are listed in Appendix A of the [Informix Guide to SQL: Reference](#).

For a complete explanation of how to create and populate the demonstration database on your database server, refer to the [DB-Access User Manual](#). ♦

For a complete explanation of how to create and populate the demonstration database, refer to the administrator's guide for your database server.

Documentation Conventions

This section describes the conventions that this manual uses. These conventions make it easier to gather information from this and other Informix manuals.

The following conventions are covered:

- Typographical conventions
- Icon conventions
- Command-line conventions
- Sample-code conventions

Typographical Conventions

This manual uses the following standard set of conventions to introduce new terms, illustrate screen displays, describe command syntax, and so forth.

Convention	Meaning
KEYWORD	All keywords appear in uppercase letters in a serif font.
<i>italics</i>	Within text, new terms and emphasized words appear in italics. Within syntax diagrams, values that you are to specify appear in italics.
boldface	Identifiers (names of classes, objects, constants, events, functions, program variables, forms, labels, and reports), environment variables, database names, filenames, table names, column names, icons, menu items, command names, and other similar terms appear in boldface.
<code>monospace</code>	Information that the product displays and information that you enter appear in a monospace typeface.
KEYSTROKE	Keys that you are to press appear in uppercase letters in a sans serif font.
◆	This symbol indicates the end of feature-, product-, platform-, or compliance-specific information.
→	This symbol indicates a menu item. For example, “Choose Tools→Options ” means choose the Options item from the Tools menu.






***Tip:** When you are instructed to “enter” characters or to “execute” a command, immediately press RETURN after the entry. When you are instructed to “type” the text or to “press” other keys, no RETURN is required.*

Icon Conventions

Throughout the documentation, you will find text that is identified by several different types of icons. This section describes these icons.




Comment Icons

Comment icons identify warnings, important notes, or tips. This information is always displayed in italics.








Icon	Description
	The <i>warning</i> icon identifies vital instructions, cautions, or critical information.
	The <i>important</i> icon identifies significant information about the feature or operation that is being described.
	The <i>tip</i> icon identifies additional details or shortcuts for the functionality that is being described.

Feature, Product, and Platform Icons

Feature, product, and platform icons identify paragraphs that contain feature-specific, product-specific, or platform-specific information.

Icon	Description
	Identifies information that is specific to an Asian Language Support (ALS) database or application.
	Identifies information that is specific to a Native Language Support (NLS) database or application.
	Identifies information that relates to the Informix Global Language Support (GLS) feature.

(1 of 2)


Icon	Description
	Identifies information that is specific to INFORMIX-ESQL/C.
	Identifies information that is specific to INFORMIX-Universal Server.
	Identifies information that is specific to INFORMIX-OnLine Dynamic Server.
	Identifies information that is specific to INFORMIX-OnLine Workgroup Server.
	Identifies information that is specific to INFORMIX-SE.
	Identifies information that is specific to the Windows NT environment.
	Identifies information that is specific to the UNIX platform.

(2 of 2)

These icons can apply to a row in a table, one or more paragraphs, or an entire section. A ♦ symbol indicates the end of the feature-, product-, or platform-specific information.

Compliance Icons

Compliance icons indicate paragraphs that provide guidelines for complying with a standard.

Icon	Description
	Identifies information that is specific to an ANSI-compliant database.

These icons can apply to a row in a table, one or more paragraphs, or an entire section. A ♦ symbol indicates the end of the compliance information.

Command-Line Conventions

This section defines and illustrates the format of commands that are available in Informix products. These commands have their own conventions, which might include alternative forms of a command, required and optional parts of the command, and so forth.

The utilities that you use to move data support a variety of command-line options. The syntax of each command is illustrated in [Chapter 10, “Utilities for Data Migration.”](#)

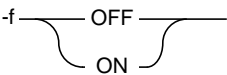
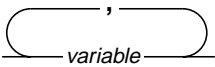
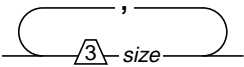
Each diagram displays the sequences of required and optional elements that are valid in a command. A diagram begins at the upper-left corner with a command. It ends at the upper-right corner with a vertical line. Between these points, you can trace any path that does not stop or back up. Each path describes a valid form of the command. You must supply a value for words that are in italics.

You might encounter one or more of the following elements on a command-line path.

Element	Description
command	This required element is usually the product name or other short word that invokes the product or calls the compiler or preprocessor script for a compiled Informix product. It might appear alone or precede one or more options. You must spell a command exactly as shown and use lowercase letters.
<i>variable</i>	A word in italics represents a value that you must supply, such as a database, file, or program name. A table following the diagram explains the value.
-flag	A flag is usually an abbreviation for a function, menu, or option name or for a compiler or preprocessor argument. You must enter a flag exactly as shown, including the preceding hyphen.
.ext	A filename extension, such as .sql or .cob , might follow a variable that represents a filename. Type this extension exactly as shown, immediately after the name of the file. The extension might be optional in certain products.

(1 of 3)

Element	Description
(.,;+* - /)	Punctuation and mathematical notations are literal symbols that you must enter exactly as shown.
' '	Single quotes are literal symbols that you must enter as shown.
<div>Privileges p. 5-17</div> <div>Privileges</div>	A reference in a box represents a subdiagram. Imagine that the subdiagram is spliced into the main diagram at this point. When a page number is not specified, the subdiagram appears on the same page.
<div>VALUES Clause see SQLS</div>	A reference to SQLS in this manual refers to the Informix Guide to SQL: Syntax . Imagine that the subdiagram is spliced into the main diagram at this point.
<div>Other options see OLAG</div>	A reference to OLAG in this manual refers to the administrator's guide for your database server. Imagine that the subdiagram is spliced into the main diagram at this point.
<div>E/C</div>	<p>An icon is a warning that this path is valid only for some products, or only under certain conditions. Characters on the icons indicate what products or conditions support the path.</p> <p>These icons might appear in a diagram:</p> <div><div>OL</div><div>This path is valid for INFORMIX-OnLine Dynamic Server, INFORMIX-Workgroup Server, and INFORMIX-Universal Server.</div></div> <div><div>SE</div><div>This path is valid only for INFORMIX-SE.</div></div>
<div>— </div>	The vertical line terminates the command.

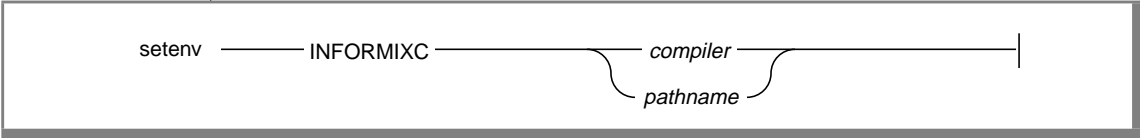
Element	Description
	A branch below the main path indicates an optional path. (Any term on the main path is required, unless a branch can circumvent it.)
	A loop indicates a path that you can repeat. Punctuation along the top of the loop indicates the separator symbol for list items.
	A gate ($\sqrt{3}$) on a path indicates that you can only use that path the indicated number of times, even if it is part of a larger loop. Here you can specify <i>size</i> no more than three times within this statement segment.

(3 of 3)

How to Read a Command-Line Diagram

Figure 1 shows a command-line diagram that uses some of the elements that are listed in the previous table.

Figure 1
Example of a Command-Line Diagram



To construct a command correctly, start at the top left with the command. Then follow the diagram to the right, including the elements that you want. The elements in the diagram are case sensitive.

Figure 1 diagrams the following steps:

1. Type the word `setenv`.
2. Type the word `INFORMIXC`.
3. Supply either a compiler name or `pathname`.
After you choose `compiler` or `pathname`, you come to the terminator. Your command is complete.
4. Press RETURN to execute the command.

Sample-Code Conventions

Examples of SQL code occur throughout this manual. Except where noted, the code is not specific to any single Informix application development tool. If only SQL statements are listed in the example, they are not delimited by semicolons. For instance, you might see the code in the following example:

```
CONNECT TO stores7
:
:
DELETE FROM customer
      WHERE customer_num = 121
:
:
COMMIT WORK
DISCONNECT CURRENT
```

To use this SQL code for a specific product, you must apply the syntax rules for that product. For example, if you are using the Query-language option of DB-Access, you must delimit multiple statements with semicolons. If you are using an SQL API, you must use EXEC SQL at the start of each statement and a semicolon (or other appropriate delimiter) at the end of the statement.

***Tip:** Ellipsis points in a code example indicate that more code would be added in a full application, but it is not necessary to show it to describe the concept being discussed.*

For detailed directions on using SQL statements for a particular application development tool or SQL API, see the manual for your product.



Additional Documentation

For additional information, you might want to refer to the following types of documentation:

- On-line manuals
- Printed manuals
- On-line help
- Error message files
- Documentation notes, release notes, and machine notes

On-Line Manuals

A CD that contains Informix manuals in electronic format is provided with your Informix products. You can install the documentation or access it directly from the CD. For information about how to install, read, and print on-line manuals, see either the installation guide for your product or the installation insert that accompanies the documentation CD.

The documentation set that is provided on the CD describes a particular Informix database server, its implementation of SQL, and its associated application-programming interfaces.

Printed Manuals

An Informix documentation set describes a particular database server, its implementation of SQL, and its associated application-programming interfaces.

To order printed manuals, call 1-800-331-1763 or send email to moreinfo@informix.com.

Please provide the following information:

- The documentation that you need
- The quantity that you need
- Your name, address, and telephone number

On-Line Help

INFORMIX-Command Center and the Relational Object Manager on-line help provide contextual information about elements on the screen and steps for carrying out various tasks. The Command Center is used to monitor and facilitate the day-to-day tasks required by a database server. ♦

ODS**OVS**

UNIX

Error Message Files

Informix software products provide ASCII files that contain all of the Informix error messages and their corrective actions. To read the error messages, Informix provides scripts that let you display error messages on the screen (**finderr**) or print formatted error messages (**rofferr**). See the Introduction to the *Informix Error Messages* manual for a detailed description of these scripts. ♦

OWS UNIX

To read the error messages on the Windows 95 client, click the **Informix Find Error** icon in the INFORMIX-OnLine Workgroup Server program group. To read the error messages on the UNIX server, use the **finderr** and **rofferr** scripts. ♦

OWS NT

OWS NT

Informix software products provide ASCII files that contain all of the Informix error messages and their corrective actions. To read the error messages, Informix provides the **Find Error** option in the Command Center. ♦

Documentation Notes, Release Notes, Machine Notes

In addition to Informix manuals, the following on-line files supplement the information in this manual.

On-Line File	Purpose
Documentation notes	This file describes features that are not covered in the manuals or that have been modified since publication. The file that contains the documentation notes for the <i>Informix Migration Guide</i> is called MIGRATEDOC_x.x where x.x is the version number of your product.
Release notes	This file describes feature differences from earlier versions of Informix products and how these differences might affect current products. This file also contains information about any known problems and their workarounds. For the name of the release-notes file for your product, see the product documentation.
IUNIVERSAL_9.1	The machine-notes file describes any special actions that are required to configure and use Informix products on your computer. Machine notes are named for the product described. ♦

UNIX

Please examine these files because they contain vital information about application and performance issues.

The following table shows the location of these on-line files for each product.

Database Server Product	Environment	Directory Name
OnLine Dynamic Server	UNIX	<code>\$INFORMIXDIR/release/en_us/0333</code>
OnLine Dynamic Server	Windows NT	<code>%INFORMIXDIR%\release</code>
OnLine Workgroup Server	UNIX	<code>\$INFORMIXDIR/release/ows</code>
OnLine Workgroup Server	Windows NT	<code>%INFORMIXDIR%\release</code>
Universal Server	UNIX	<code>\$INFORMIXDIR/release/en_us/0333</code>

Compliance with Industry Standards

The American National Standards Institute (ANSI) has established a set of industry standards for SQL. Informix SQL-based products are fully compliant with SQL-92 Entry Level (published as ANSI X3.135-1992), which is identical to ISO 9075:1992 on INFORMIX-OnLine Dynamic Server, INFORMIX-OnLine Workgroup Server, and INFORMIX-Universal Server. In addition, many features of OnLine Dynamic Server, OnLine Workgroup Server, and Universal Server comply with the SQL-92 Intermediate and Full Level and X/Open SQL CAE (common applications environment) standards.

Informix SQL-based products are compliant with ANSI SQL-92 Entry Level (published as ANSI X3.135-1992) on INFORMIX-SE with the following exceptions:

- Effective checking of constraints
- Serializable transactions

Informix Welcomes Your Comments

Please tell us what you like or dislike about our manuals. To help us with future versions of our manuals, we want to know about corrections or clarifications that you would find useful. Include the following information:

- The name and version of the manual that you are using
- Any comments that you have about the manual
- Your name, address, and phone number

Write to us at the following address:

Informix Software, Inc.
SCT Technical Publications Department
4100 Bohannon Drive
Menlo Park, CA 94025

If you prefer to send email, our address is:

`doc@informix.com`

Or send a facsimile to the Informix Technical Publications Department at:

415-926-6571

We appreciate your feedback.

Migrating Data

Section I



Data Migration Overview

Planning for Data Migration	1-3
Migrating Between Versions	1-3
Moving to a Different Database Server or Environment	1-4
Migrating Client Applications.	1-4
Importing Data from Other Sources.	1-4
Overview of Informix Database Servers.	1-5
Where to Find Information on Migrating to a Database Server	1-6
Upgrading Database Servers	1-6
Reverting to an Earlier Version	1-7
Migrating Between Database Servers in the Same Environment	1-7
Migrating to a Different Database Server Environment	1-8
Tools for Moving Data.	1-9
Choosing the Appropriate Migration Utility.	1-11
Using the High-Performance Loader	1-11
Using the onunload and onload Utilities.	1-11
Choosing Between dbload, dbimport, and LOAD	1-13

This chapter discusses the reasons that you might have for data migration, includes a matrix of the migration paths, and introduces the tools that you can use for migration.

Planning for Data Migration

You might need to perform data migration on a database or selected data for the following reasons:

- To migrate from one version of a database server to another version
- To move to a different database server or environment
- To distribute an application to users
- To import data generated from a source other than INFORMIX-Universal Server, INFORMIX-OnLine Dynamic Server, INFORMIX-OnLine Workgroup Server, or INFORMIX-SE

Migrating Between Versions

When you migrate between versions of a database server (for example, from OnLine Dynamic Server, Version 7.2x to Universal Server), you need to consider changes in the configuration parameters and environment variables, the amount of memory and dbspace space required, and the organization of the data.

Moving to a Different Database Server or Environment

When you migrate to a different environment, you might be moving between database servers, moving to a different operating system, or moving to a different locale. If you are moving to a different operating system or INFORMIX-SE, you must unload the data into text files and then load it back again with the target database server.

You also need to review the organization of the data. You might want to change the database schema to accommodate more information, to provide for growth, or to make the organization more convenient.

Migrating Client Applications

When you migrate from one environment to another, consider whether to also migrate the client applications that access the data. Regardless of whether you migrate the client applications, you must update the **sqlhosts** information for the client applications with the new database server information.

Client applications should be the same version as the database server.

After you upgrade a database server on the same operating-system platform, or move the database server to another compatible computer, review the client applications and **sqlhosts** connections. You might need to recompile or modify the client applications, or update the **sqlhosts** information. For more information about interactions between client applications and different database servers, refer to the application manual, such as the [INFORMIX-ESQL/C Programmer's Manual](#) or the [DataBlade API Programmer's Manual](#).

Importing Data from Other Sources

You can use the **dbimport** and **dbload** utilities, the High-Performance Loader (HPL), or the INFORMIX-Gateway products to import data from non-Informix sources. See [“Importing Data from a Non-Informix Environment” on page 2-8](#) for more information.

Overview of Informix Database Servers

If you are unfamiliar with the Informix client/server environment, read *Getting Started with Informix Database Server Products*. It discusses the differences between OnLine Dynamic Server and SE, and network and server configuration. For information on installing and configuring Universal Server, and on the new features, read [Getting Started with INFORMIX-Universal Server](#).

Several types of database servers are available from Informix. When you choose your database server, you consider the size of your databases and whether they contain simple or complex data objects, speed, the need for distributed access, hardware platforms, scalability, and ease of installation and maintenance. Unique SQL features are available on each type of database server. If you create a database with one database server, later you can migrate the data in the database to another database server. Figure 1-1 lists the Informix database servers by name, latest version level, and the environments they support.

Figure 1-1
Informix Database Servers

Server Type	Version Level	Environments Supported
INFORMIX-Universal Server	9.1	UNIX, Windows NT
INFORMIX-OnLine Dynamic Server	7.22	UNIX, Windows NT
INFORMIX-OnLine Workgroup Server	7.22	Windows NT
INFORMIX-OnLine Workgroup Server	7.20	UNIX
INFORMIX-SE	7.22	UNIX

After you choose a database server, read the installation guide for your database server version for information on installing the database server. Read the administrator's guide for information on configuring and operating your database server.

Where to Find Information on Migrating to a Database Server

This section summarizes all of the migration paths for Informix database servers. The Source column shows the original database server and environment. The Target column shows the new database server and environment to which you are migrating. The Page Reference column shows where you can find more information about the specific migration path.

Upgrading Database Servers

Figure 1-2 shows where you can find information on upgrading database servers. For example, if you plan to upgrade to Universal Server, refer to [“Migrating from OnLine Dynamic Server 7.2 to Universal Server”](#) on page 3-6.

Figure 1-2
Upgrading Database Servers

Source	Target	Page Reference
ODS/UNIX 7.2	Universal Server	3-6
ODS/UNIX 6.0	ODS/UNIX 7.2x	4-19
ODS/UNIX (pre-6.0)	ODS/UNIX 7.2x	5-24
ODS/NT 7.12	ODS/NT 7.22	4-26
OWS/NT 7.12	OWS/NT 7.22	6-9
OWS/UNIX 7.12	OWS/UNIX 7.20	6-9
SE (any earlier version)	SE 7.2x	7-4

Reverting to an Earlier Version

Figure 1-3 shows where you can find information on reverting to an earlier version of the database server.

Figure 1-3
Reverting to an Earlier Version

Source	Target	Page Reference
Universal Server	ODS/UNIX 7.2	3-20
ODS/UNIX 7.2x	ODS/UNIX 6.0 or later	4-33
ODS/UNIX 7.2x	ODS/UNIX (pre-6.0)	5-39
ODS/NT 7.22	ODS/NT 7.12 or 7.2	4-44
OWS/NT 7.22	OWS/NT 7.12	6-17
OWS/UNIX 7.20	OWS/UNIX 7.12	6-17
SE 7.2x	SE (any earlier version)	7-6

Migrating Between Database Servers in the Same Environment

[Figure 1-4 on page 1-8](#) shows situations where data migration is automatic. Automatic data migration means you do not need to use **dbexport**, **UNLOAD**, or **onunload** to move the data. Simply install and start the new server. For example, data migration between OnLine Dynamic Server (ODS) and OnLine Workgroup Server (OWS) is automatic if they are in the same environment. Data migration is also automatic when you move between different versions of a database server.

Figure 1-4
Migrating Between Different Database Servers in the Same Environment

Source	Target	Page Reference
ODS/NT	OWS/NT	8-17
ODS/UNIX	OWS/UNIX	8-17
OWS/NT	ODS/NT	8-26
OWS/UNIX	ODS/UNIX	8-26

Migrating to a Different Database Server Environment

Figure 1-5 shows all the paths for migrating to a different database server environment. For example, if you plan to migrate from SE to OnLine Workgroup Server, refer to [“Moving Data from SE to OnLine Dynamic Server or OnLine Workgroup Server”](#) on [pages 8-6 through 8-11](#).

Figure 1-5
Migrating to a Different Database Server Environment

Source	Target	Page Reference
C-ISAM	SE, ODS, or OWS	7-6
ODS/NT	ODS/UNIX	8-36
ODS/NT	OWS/UNIX	8-17
ODS/NT or UNIX	SE (all environments)	8-12
ODS/UNIX	ODS/NT	8-36
ODS/UNIX	OWS/NT	8-17
OWS/NT	ODS/UNIX	8-27
OWS/NT	OWS/UNIX	8-38
OWS/NT or UNIX	SE (all environments)	8-12
OWS/UNIX	ODS/NT	8-27

(1 of 2)

Source	Target	Page Reference
OWS/UNIX	OWS/NT	8-38
SE (all environments)	ODS/NT or UNIX	8-6
SE (all environments)	OWS/NT or UNIX	8-6

(2 of 2)

Tools for Moving Data

Informix provides several utilities and SQL statements for moving data from one database to another. All Informix database servers support these utilities. This section lists these utilities, gives a brief description of each utility, and gives guidelines for choosing the correct migration utility (see [Chapter 10](#), “Utilities for Data Migration”):

- The **onunload** and **onload** utilities
- The **dbexport** and **dbimport** utilities
- The LOAD and UNLOAD statements
- The **dbload** utility
- High-Performance Loader (HPL) ♦

The best method for moving your data depends on your environment and the type of data that you want to move (a database, selected tables, or selected columns from a table). Figure 1-6 summarizes the characteristics of the commands and utilities that load information into a database.

Figure 1-6
Comparison of Tools for Loading Data

	onload/onunload	dbexport/dbimport	UNLOAD/LOAD	dbload	HPL
Granularity of data	Table or database	Database only	Partial or complete table	Partial or complete table	Partial or complete table
Performance	Fast	Moderate	Moderate	Slow	Very fast

(1 of 2)

ODS, IUS

	onload/onunload	dbexport/dbimport	UNLOAD/LOAD	dbload	HPL
Source of data	Must be produced by onunload	Usually produced by dbexport	Any data in the specified format, usually produced by UNLOAD	Any data in the format specified by the input file	Any ASCII or COBOL data. User can create custom read capabilities.
Database schema	Cannot modify	Can modify	Can modify	Can modify	Can modify
Location of data	Disk or tape	Disk or tape	Disk only	Disk only	Disk, tape, or pipe
Type of file	Binary	Text	Text	Text	Text
Logging status	Logging must be turned off	Logging optional	Logging optional	Logging optional	Logging optional
Environment	Cannot use when you move between environments	Can use when you move between environments	Can use when you move between environments	Can use when you move between environments	Can use when you move between environments or from a non-Informix database
Ease of use	More difficult	Moderate	Easiest	Moderate	Most difficult

(2 of 2)

Each of the methods for moving Informix database server data imposes constraints of one form or another on the user. The **onunload/onload** utilities provide the fastest way to move data, but you cannot modify the database schema or use **onunload** and **onload** to move from one hardware platform to another. The **dbexport/dbimport** utilities provide some flexibility, but you must move an entire database. The **dbload** utility gives you a great deal of flexibility, but it is not as fast as the other methods, and you must prepare a command file to control the input. You can use **dbload** with data in a variety of formats. The LOAD statement is moderately fast and easy to use, but it can only accept specified data formats. You usually use LOAD with data that was prepared with an UNLOAD statement.

Choosing the Appropriate Migration Utility

This section provides guidelines for choosing the appropriate migration utility.

Using the High-Performance Loader

High-Performance Loader (HPL) requires significant preparation time but is fast. Use the HPL for large migration jobs. The HPL can load data from any ASCII or COBOL file that meets certain format requirements. For information about how to use the HPL, refer to the [Guide to the High-Performance Loader](#).

Important: Only OnLine Dynamic Server for UNIX, Version 7.2x, and Universal Server support the HPL. OnLine Dynamic Server for Windows NT and OnLine Workgroup Server do not support the HPL. ♦

Using the onunload and onload Utilities

The **onunload** utility unloads data from the specified database or table onto a tape or a file on disk in disk-page-sized units. The **onload** utility takes a tape or a file created by the **onunload** utility and re-creates the database or the table. The **onunload** and **onload** utilities are faster than the **dbimport**, **dbload**, or **LOAD** options but are much less flexible.

Because the data is written in page-sized units, you can use **onunload** and **onload** only when certain constraints are met. For example, you cannot use **onunload** and **onload** to move data between UNIX and Windows NT systems. These constraints are discussed in [“Constraints That Affect onunload” on page 10-69](#). [Figure 1-7 on page 1-12](#) summarizes the questions that you must ask before you use **onunload** and **onload**.

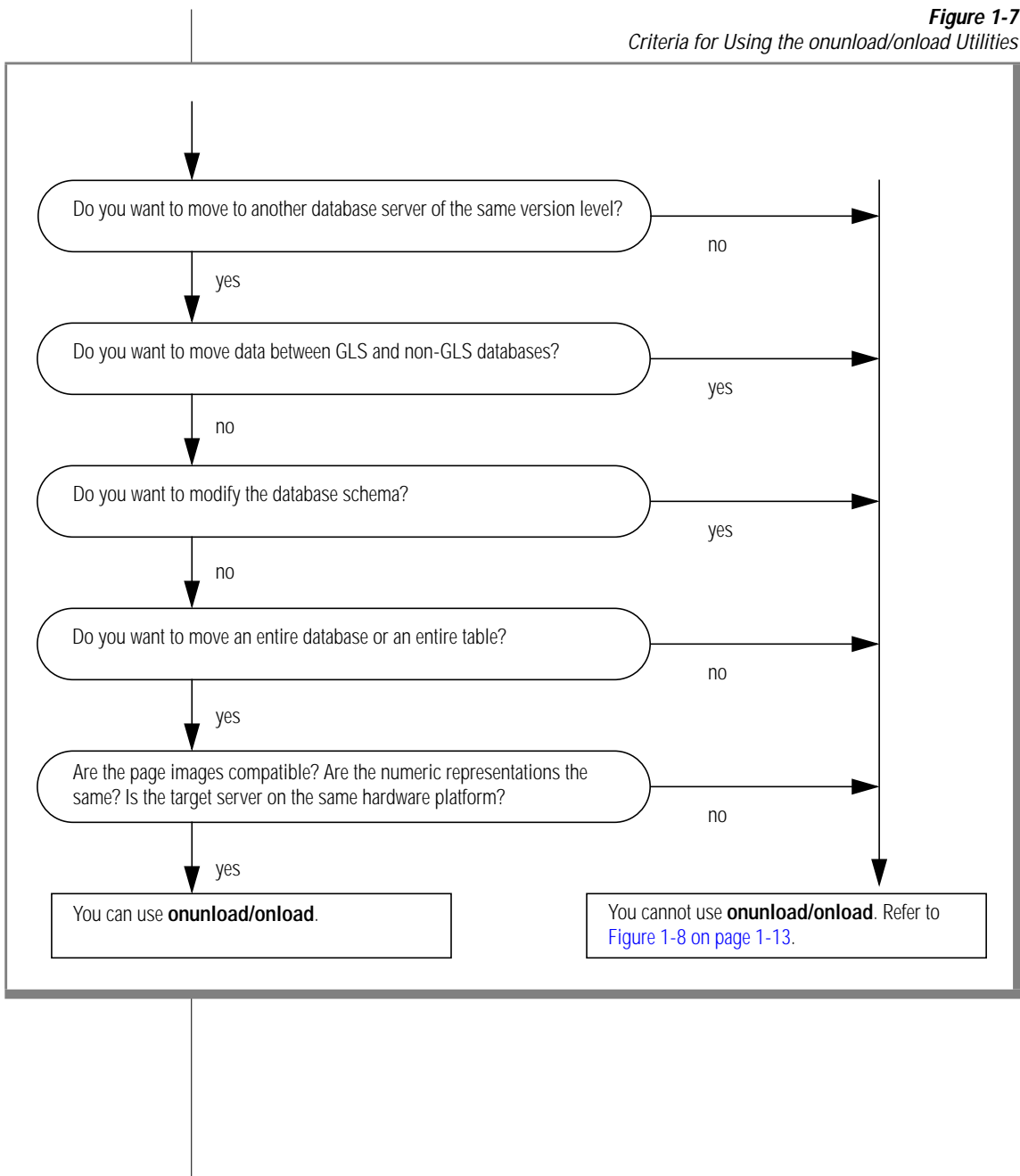
IUS

ODS

UNIX



Figure 1-7
Criteria for Using the *onunload/onload* Utilities



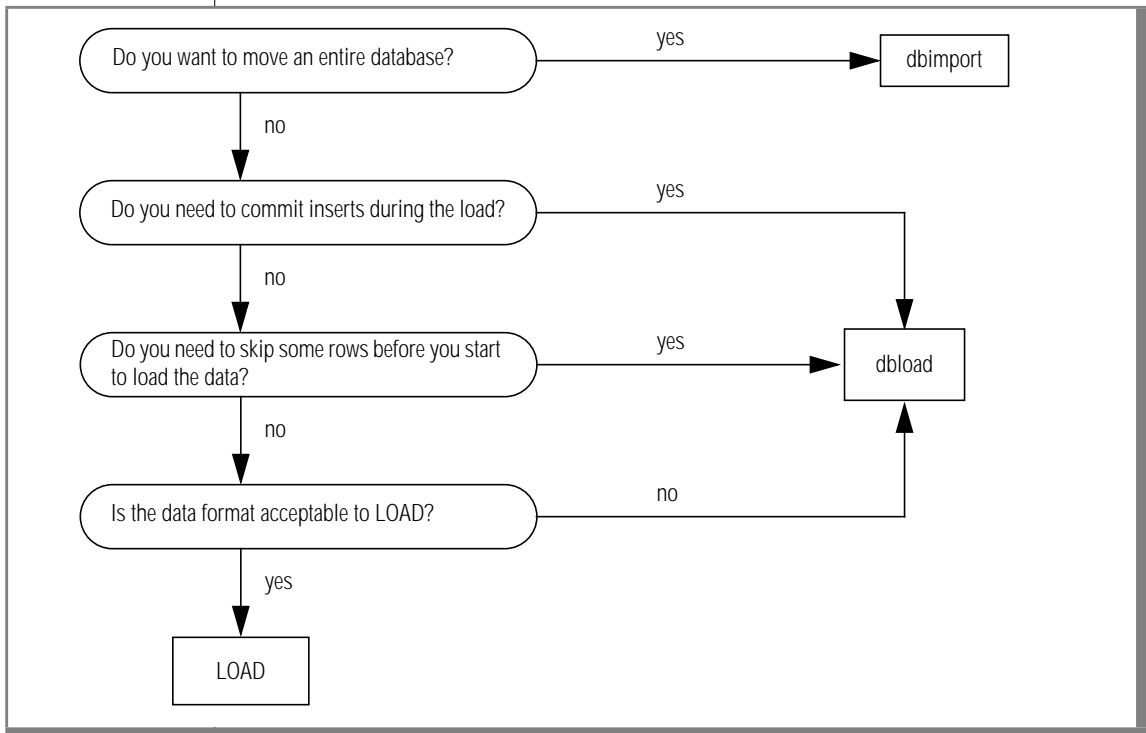
Choosing Between **dbload**, **dbimport**, and **LOAD**

If you cannot use **onunload** and **onload**, you must choose between the **dbload** and **dbexport/dbimport** utilities and the **LOAD**/**UNLOAD** statements. All these methods let you modify the database schema.

The **dbimport** utility loads a complete database. You use **dbimport** when you must move a complete database and cannot use **onload**. To load tables, use **LOAD** or **dbload**.

If you must manipulate the data file or access the database while it is unloading, use the **dbload** utility. The cost of the flexibility is time spent creating the **dbload** command file and slower execution. When possible, most users prefer to use the **LOAD** statement because of its simplicity. The **LOAD** statement also is faster than **dbload**. Figure 1-8 summarizes the questions that you must ask to choose among these methods.

Figure 1-8
Choosing Among **dbimport**, **dbload**, and **LOAD**



Moving Between Computers

Moving Database Server Data Between Computers Using onunload and onload	2-3
Steps for Using onunload and onload	2-4
Moving Data to a New Environment.	2-7
Importing Data from a Non-Informix Environment	2-8
Using the Informix Gateway Products	2-8
Using the High-Performance Loader	2-9

This chapter discusses moving data between different computers and importing data from non-Informix environments. Except when you use the High-Performance Loader (HPL), you must unload your data to ASCII files before you move the data to another computer.

Moving Database Server Data Between Computers Using **onunload** and **onload**

When you move database server data between compatible computers, you can use the **onunload** and **onload** utilities to unload and load the data, as follows:

- OnLine Dynamic Server data between UNIX computers
- OnLine Dynamic Server data between Windows NT computers
- OnLine Workgroup Server data between UNIX computers
- OnLine Workgroup Server data between Windows NT computers
- Universal Server data between UNIX computers (only if the database does not contain extended data types)

For example, your site purchased a more powerful UNIX computer to allow faster access to the users. You need to transfer existing databases to the new database server on the new computer. Use **onunload** to unload data from the first database server; then use **onload** to load the data into the second database server. Both database servers must be at the same version level. You can move the entire database or selected tables only, but you cannot modify the database schema.



IUS

SE

The **onunload** and **onload** utilities are the fastest way to unload and load data, but you can use them only when the source and target computers have the same page size, use the same representation of numeric data, and have the same byte alignment for structures and unions. You cannot use **onunload** and **onload** to move data between UNIX and Windows NT computers because they have different page sizes. For example, the page size is 2 kilobytes on some UNIX systems and 4 kilobytes on Windows NT.

Important: The **onunload** and **onload** utilities have the following restrictions:

- You cannot use **onunload** and **onload** to move data between non-GLS and GLS locales.
- Do not use **onunload** and **onload** to move data between two Universal Server databases if they contain extended data types. Use the HPL instead to move the Universal Server data. However, you can use **onunload** and **onload** with Universal Server if the databases contain legacy data types. ♦
- INFORMIX-SE does not support **onload** and **onunload**. ♦

Steps for Using *onunload* and *onload*

This section describes the procedure for how to use **onunload** and **onload**. You can use these commands to move either a complete database or a table from one computer to another. The syntax and description of the **onunload** utility starts on [page 10-68](#). The syntax and description of the **onload** utility starts on [page 10-56](#).

To move a database from one computer to another

1. Make sure that the page size, numeric representations, and byte alignment on structures and unions are the same on both computers. (The page size is 2 kilobytes on certain UNIX systems and 4 kilobytes on Windows NT.)

The page size is an Informix characteristic. For information about page size, refer to the administrator's guide for your database server. The numeric representation and the byte alignment are characteristics of your operating system. Refer to the manuals for your operating systems.

2. Decide where to store the unloaded data.
 - On disk Create an empty file for **onunload** to hold the data. Make sure that you have write permission for the file.
 - On tape Use the tape device and characteristics specified in the ONCONFIG configuration file by either TAPEDEV or LTAPEDEV, or specify another tape device. Make sure that the tape device that you specify is available for use by **onunload**.
3. Run the **oncheck** utility to make sure that your database is consistent.

For information about **oncheck**, refer to the administrator's guide for your database server product.
4. If you wish to save the triggers, access privileges, stored procedures, defaults, constraints, and synonyms for the tables in the database, run the **dbschema** utility.
5. Run the **onunload** utility.
6. If necessary, transfer the storage medium (tape or disk) to the new computer. If the two computers are on the same network, you can read or write the data remotely.
7. Run the **onload** utility.
8. Set the desired logging status for the new database. For information about logging status, refer to the administrator's guide for your database server product.
9. If necessary, change the DBA privileges of the database.
10. If you wish to restore the triggers, access privileges, stored procedures, defaults, constraints, and synonyms for the tables in the database, run the **dbschema** utility.
11. Create a level-0 backup of the new database.

To move a table from one computer to another

1. Make sure that the page size, numeric representations, and byte alignment on structures and unions are the same on both computers. (The page size is 2 kilobytes on certain UNIX systems and 4 kilobytes on Windows NT.)
2. Decide where to store the unloaded data. (See step 2 of the previous section.)
3. Run the **oncheck** utility to make sure that your database is consistent.
4. If you wish to save the triggers, access privileges, stored procedures, defaults, constraints, and synonyms for the table, run the **dbschema** utility.
5. Run the **onunload** utility.
6. If necessary, transfer the storage medium to the new computer.
7. If the table includes simple large objects that are stored in blobspaces, decide where to store the simple large objects. If necessary, create new blobspaces.
8. Turn off logging.

When you are loading a table, logging on the target database must be turned off. (When you are creating and loading an entire database, the logging status does not matter.)
9. Run the **onload** utility.
10. Create a level-0 backup of the modified database.
11. Turn logging back on, if desired.
12. If you wish to restore the triggers, access privileges, stored procedures, defaults, constraints, and synonyms for the table, run the **dbschema** utility or create them manually.

To move a table from one dbspace to another dbspace on the same computer

1. Run the **onunload** utility to unload the table.
2. Turn off logging. When you are loading a table, logging on the target database must be turned off.
3. Run the **onload** utility. Specify a new table name and new dbspace name in the **onload** statement.
4. If the data loads successfully, drop the old table in the old dbspace and rename the new table to the old table name.
5. Create a level-0 backup of the modified database.
6. Turn logging back on, if desired.

Moving Data to a New Environment

If you cannot use the **onunload** and **onload** utilities to export and import data, you must unload your data to text files. You can use the **dbexport** utility to unload Universal Server, OnLine Dynamic Server, OnLine Workgroup Server, and SE data to tape. The UNLOAD statement lets you manipulate the data as you unload it, but it requires that you unload to files on disk instead of to tape. If you unload to files, you might need to use UNIX or Windows NT utilities to load those files onto tape.

If you are moving to an Informix database server on another computer, you can use the **dbimport** and **dbload** utilities to load the data that you exported.

If you are moving data to a non-Informix application, you might need to use the UNLOAD statement because it lets you specify the delimiter that is used in the data files.

For more information on the **dbexport**, **dbimport**, **dbload**, and **dbschema** utilities, see [Chapter 10, “Utilities for Data Migration.”](#) For more information about the UNLOAD and LOAD statements, refer to the [Informix Guide to SQL: Syntax](#).

Importing Data from a Non-Informix Environment

The **dbimport** and **dbload** utilities can import data from any ASCII file that is properly formatted. Most applications that produce data can export the data into files that have a suitable format for **dbimport**. If not, you must use UNIX or Windows NT utilities to reformat the data before you import it into Universal Server, OnLine Dynamic Server, OnLine Workgroup Server, or SE. In addition to **dbimport** and **dbload**, the Informix Gateway products and the High-Performance Loader (HPL) provide ways to access information from non-Informix sources.

Using the Informix Gateway Products

INFORMIX-Enterprise Gateway *with DRDA* lets you query databases that conform to the DRDA protocol published by IBM. You can use this Gateway product to query a DRDA database and then insert the results into an Informix database. For example, to import data, execute a SELECT statement from the non-Informix database and then an INSERT statement into the Informix database. For more information about INFORMIX-Enterprise Gateway *with DRDA*, refer to the *INFORMIX-Enterprise Gateway with DRDA User Manual*.

INFORMIX-Enterprise Gateway *for EDA/SQL* lets you issue queries on a variety of hardware platforms. It accesses an EDA/SQL Server from Information Builders, Inc. (IBI), which in turn accesses the data source. For more information about INFORMIX-Enterprise Gateway *for EDA/SQL*, refer to the *INFORMIX-Enterprise Gateway for EDA/SQL User Manual*.

INFORMIX-Enterprise Gateway Manager provides a single, standards-based gateway to multiple data sources. Gateway Manager connects the Informix environment with that of any shared-library ODBC Level 2-compliant driver manager and driver(s) on UNIX. For instance, you can use Gateway Manager with the INFORMIX-Enterprise Gateway driver products to access UNIX database server products such as SYBASE SQL Server 10 and ORACLE7 Server. For more information about Gateway Manager, refer to the *INFORMIX-Enterprise Gateway Manager User Manual*.

IUS

ODS

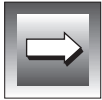
UNIX

Using the High-Performance Loader

The High-Performance Loader (HPL) is a utility available with OnLine Dynamic Server and Universal Server, that uses parallel processing to perform fast data loading and unloading. You can use the HPL to load from large ASCII or COBOL databases. In addition to the advantage of its speed, the following HPL features provide powerful tools for handling data from non-Informix sources:

- Drivers to handle different database types
- Filters and functions to manipulate data
- Code-set conversion

For more information about the HPL, refer to the [Guide to the High-Performance Loader](#). ♦



Important: Only Universal Server and OnLine Dynamic Server for UNIX, Version 7.2x, support the HPL. OnLine Dynamic Server for Windows NT and OnLine Workgroup Server for Windows NT and UNIX do not support the HPL.

Upgrading Versions of Database Servers

Section II



Upgrading OnLine Dynamic Server 7.2 to Universal Server

Preparing to Migrate to Universal Server	3-3
Changes Introduced by Universal Server	3-4
Environment Variables	3-4
Configuration Parameters	3-5
New Database Storage Spaces.	3-6
Global Language Support	3-6
Migrating from OnLine Dynamic Server 7.2 to Universal Server	3-6
Check Available Space	3-8
Save Copies of the Current Configuration Files.	3-9
Close All Transactions	3-10
Initiate a Fast Recovery	3-10
Verify the Integrity of the Data	3-11
Verify the Mode	3-12
Make a Final Backup	3-12
Bring OnLine Dynamic Server Off-Line	3-13
Reconfigure the UNIX Operating System	3-14
Install Universal Server.	3-14
Verify That Environment Variables Are Set Correctly.	3-15
Update the ONCONFIG Configuration Files.	3-15
Add a Communications Support Module (Optional)	3-16
Using Communications Support Modules with Client Applications.	3-16
Update the ON-Archive Configuration Files.	3-16
Bring Universal Server On-Line	3-17
Verify the Integrity of the Data	3-18
Make an Initial Universal Server Backup	3-18
Migration Complete	3-19
Install and Configure DataBlade Modules	3-19

Reverting Universal Server to OnLine Dynamic Server 7.2	3-20
Determine Whether Reversion is Possible	3-21
Save Copies of the Configuration Files	3-21
Verify the Integrity of the Data	3-22
Back Up the Universal Server Data	3-22
Remove Universal Server Features	3-22
Uninstall DataBlade Modules	3-23
Remove User-Defined Routines and User-Defined Functions.	3-23
Drop Indexes Over 255 Bytes	3-24
Remove Extended Data Types and Smart Large Objects	3-24
Remove Universal Server-Specific Features	3-25
Remove Stored Procedures Created in Universal Server	3-25
Drop Secondary Access Methods (Indexes)	3-25
Remove Virtual Tables in External Spaces	3-26
Remove Sbspaces and External Spaces	3-26
Run the Reversion Utility	3-26
Modify Configuration Parameters	3-27
Reset Environment Variables	3-27
Remove Communications Support Module Settings	3-28
Install and Start OnLine Dynamic Server	3-28
Verify the Integrity of the Data	3-28
Back Up OnLine Dynamic Server Data	3-28
Return OnLine Dynamic Server to On-Line Mode	3-28

T

his chapter describes the procedure for migrating between OnLine Dynamic Server, Version 7.2, and INFORMIX-Universal Server, on the UNIX operating system.

This chapter covers the following topics:

- Upgrading OnLine Dynamic Server 7.2 to Universal Server
- Reverting Universal Server to OnLine Dynamic Server 7.2



Important: *If you want to migrate to Universal Server from an earlier release of INFORMIX-OnLine or OnLine Dynamic Server, you must first migrate to OnLine Dynamic Server, Version 7.2, and then follow the instructions in this chapter to migrate to Universal Server. For instructions on how to migrate to OnLine Dynamic Server, Version 7.2x, from an earlier release of OnLine or OnLine Dynamic Server, refer to [Chapter 4](#) and [Chapter 5](#). (Version 7.2x refers to versions 7.2, 7.21, or 7.22.)*

Preparing to Migrate to Universal Server

Informix suggests that you observe the following precautions when you migrate to Universal Server:

- Check the release notes for information about the proper operating-system release and any patches that are required for successful installation and operation of the database server.

The release notes are in the `$INFORMIXDIR/release/en_us/0333` directory on UNIX.

- Retain both versions of the Informix product software on disk (if you have enough disk resources).
- Retain the installation tapes from both versions of the Informix product software.

- Perform a level-0 back up of all dbspaces and blobspaces with OnLine Dynamic Server, Version 7.2. After you complete the migration, perform another level-0 back up with Universal Server.
- Use a test instance of Universal Server to test the installation and migration procedures. Use a test instance in the desired communication mode to practice bringing the new database server on-line before you attempt to convert the production database.

For additional information, refer to both the [INFORMIX-Universal Server Installation Guide](#) and [Getting Started with INFORMIX-Universal Server](#).

Changes Introduced by Universal Server

This section describes the changes introduced by Universal Server that affect either migration or initial configuration.

Environment Variables

Universal Server introduces the following new environment variables that were not available in OnLine Dynamic Server, Version 7.2:

- **INFORMIXCONCSMCFG**
- **INFORMIXKEYTAB**

Review the descriptions of these environment variables for Communications Support Services to determine whether you need to set them. The [Informix Guide to SQL: Reference](#) manual describes these environment variables.

Configuration Parameters

Universal Server introduces new configuration parameters that might affect your installation. You might also need to adjust the values of existing parameters. The following new configuration parameters for Universal Server are described in the [INFORMIX-Universal Server Administrator's Guide](#):

- HETERO_COMMIT
- SBSPACENAME
- VPCLASS

The SBSPACENAME parameter specifies a default *sbspace* for the storage of *smart large objects*. For more information on smart large objects and sbspaces, refer to “[New Database Storage Spaces](#)” and to the [INFORMIX-Universal Server Administrator's Guide](#).

The VPCLASS parameter combines options from several virtual-processor configuration parameters and enables you to configure a virtual processor class with a single parameter. Consequently, if you use the VPCLASS parameter, you should be aware that it affects the following parameters, which you may need to change or delete:

- AFF_NPROCS
- AFF_SPROC
- NOAGE
- MULTIPROCESSOR
- NUMAIOVPS
- NUMCPUVPS
- SINGLE_CPU_VP

New Database Storage Spaces

Universal Server introduces the following two new database storage spaces:

- Sbspaces
- External spaces

An sbspace provides storage space for smart large objects. An external space is a space outside of the direct control of Universal Server, and one which contains data that you want to include in a database. If you use external spaces, you must also define the access methods that Universal Server will use to retrieve the external data. For more information on external spaces, refer to the [INFORMIX-Universal Server Administrator's Guide](#). For more information on user-defined access methods, refer to the *Virtual-Table Interface Programmer's Manual*.

Global Language Support

GLS

INFORMIX-Universal Server incorporates Global Language Support (GLS). GLS lets Universal Server handle different languages, cultural conventions, and code sets for Asian, European, Latin American, and Middle Eastern countries. The [Guide to GLS Functionality](#) provides a full description of GLS. ♦

Migrating from OnLine Dynamic Server 7.2 to Universal Server

This section describes the migration procedure from OnLine Dynamic Server, Version 7.2, to Universal Server, on UNIX.

When you move from OnLine Dynamic Server, Version 7.2, to Universal Server, you can install and test Universal Server with the same configuration files, environment variables, and **sqlhosts** information that you used for OnLine Dynamic Server. After you install Universal Server and verify that it works, you might want to modify configuration files and environment variables to take advantage of the Universal Server features. For more information, refer to [Getting Started with INFORMIX-Universal Server](#) and the [INFORMIX-Universal Server Administrator's Guide](#).

When you migrate to Universal Server, you need to complete the following steps, which are described in more detail in the sections that follow:

1. Check available space.
2. Save copies of the current configuration files.
3. Close all transactions with OnLine Dynamic Server, Version 7.2.
4. Initiate a fast recovery.
5. Verify the integrity of the data.
6. Verify the mode.
7. Make a final (level-0) backup.
8. Bring the current OnLine Dynamic Server off-line.

Important: Repeat steps 2 through 8 for each instance of OnLine Dynamic Server 7.2 that you are converting to Universal Server.

9. Reconfigure the operating system, if necessary.
10. Install Universal Server.
11. Verify that environment variables are set correctly.
12. Update the ONCONFIG configuration files.
13. Update the backup and restore configuration parameters.
14. Bring Universal Server on-line.
15. Verify the integrity of the data.
16. Make an initial (level-0) backup under Universal Server.

Important: Repeat steps 12 through 16 for each instance of Universal Server that you run on the computer.

17. Install and configure any DataBlade modules that you are adding to Universal Server.



Check Available Space

Universal Server requires 3000 free pages of logical-log space (approximately 6000 kilobytes for a 2-kilobyte page size) to build the **sysmaster** database.

Universal Server requires approximately 2000 kilobytes more space per database than OnLine Dynamic Server 7.2. The extra space is used for new system catalogs and built-in functions that support the extensibility features of Universal Server.

When you initialize Universal Server on an existing OnLine Dynamic Server root dbspace, it automatically upgrades the **sysmaster** database. Each database is individually upgraded when it is first accessed. To upgrade each database successfully, you must ensure that the 2000 kilobytes per database is available in each dbspace.

1. Calculate the amount of free space required for each dbspace, where n is the number of databases in the dbspace and X is the amount of free space required:

$$X \text{ kilobytes free space} = 2000 \text{ kilobytes} * n$$

2. Check the amount of free space in each dbspace to determine whether you need to add more space.

Use the following SQL statements to determine the free space required and the free space available. These statements return the free-space calculation in page-size units. The variable **free_space_req** displays the free space required and the variable **free_space_avail** displays the free space available.

To determine the free space required for each dbspace:

```
DATABASE sysmaster;
SELECT partdbsnum(partnum) dbspace_num,
       trunc(count(*) * 2000) free_space_req
FROM sysdatabases
GROUP BY 1
ORDER BY 1;
```

The following SQL statement queries the **syschunks** table and displays the free space available for each dbspace:

```
SELECT dbsnum dbspace_num, sum(nfree) free_space_avail
FROM syschunks
GROUP BY 1
ORDER BY 1;
```



Important: If a dbspace has less free space available than required, you should either move a table from it to another dbspace or add a chunk to it.

The dbspace estimates could be higher if you have an unusually large number of stored procedures in the database or an unusually large number of indexes.

Save Copies of the Current Configuration Files

Save copies of the configuration files for each instance of OnLine Dynamic Server. Keep the copies available for later use. Save the following files (if they are present):

- \$INFORMIXDIR/etc/\$ONCONFIG
- \$INFORMIXDIR/etc/onconfig
- \$INFORMIXDIR/etc/onconfig.std
- \$INFORMIXDIR\$/aaodir/adtcfg.*
- \$INFORMIXDIR\$/dbssodir/adtmasks.*
- \$INFORMIXDIR/etc/sqlhosts
- \$INFORMIXDIR/etc/tctermcap
- \$INFORMIXDIR/etc/termcap

If you use ON-Archive to back up and restore your OnLine Dynamic Server databases and the logical log, you must also copy and save the following files:

- \$INFORMIXDIR/etc/\$ARC_CONFIG
- \$INFORMIXDIR/etc/config.arc
- \$INFORMIXDIR/etc/oper_deflt.arc

Close All Transactions

Communicate to client users how long you expect the database server to be off-line for the migration. Terminate all database server processes, and shut down OnLine Dynamic Server.

To shut down the system gracefully

To shut down OnLine Dynamic Server in two steps and give users a chance to exit:

1. Execute the **onmode -sy** command.
2. Wait for all users to exit.
3. Execute the **onmode -ky** command.

To perform an immediate shutdown

To shut down OnLine Dynamic Server immediately, execute the following command:

```
onmode -ky
```

Initiate a Fast Recovery

Execute the following command to enter quiescent mode and initiate a fast recovery of your current database:

```
oninit -s
```

The **oninit -s** option rolls forward all committed transactions and rolls back all incomplete transactions since the last checkpoint and then leaves a new checkpoint record in the log with no open transactions pending. (For more information about fast recovery, refer to the *INFORMIX-OnLine Dynamic Server Administrator's Guide*.)

You must execute **oninit -s** before you initialize Universal Server. If the system is not left in a quiescent state, you get the following error when you attempt to initialize Universal Server and Universal Server goes off-line:

```
Open transaction detected when changing log versions.
```

Verify the Integrity of the Data

Use the **oncheck** utility to verify the integrity of data before you make a level-0 backup. If you find any problems with the data, fix them before you make the backup. You can verify the integrity of the reserve pages, extents, system catalog tables, data, and indexes.

To obtain a list of the databases on your database server

To get the database names, use DB-Access as follows:

```
DATABASE sysmaster;  
SELECT name FROM sysdatabases;
```

Figure 3-1 lists the commands for verifying data integrity.

Action	oncheck Command
Check reserve pages	oncheck -cr
Check extents	oncheck -ce
Check system catalog tables	oncheck -cc <i>database_name</i>
Check data	oncheck -cD <i>database_name</i>
Check indexes	oncheck -cI <i>database_name</i>

Figure 3-1
*Commands for
Verifying the Data
Integrity*

For information on **oncheck**, refer to the *INFORMIX-OnLine Dynamic Server Administrator's Guide*.

Verify the Mode

Before you make a backup, execute the following command to verify that OnLine Dynamic Server is in quiescent mode:

```
onstat -
```

The first line of the **onstat** output contains the status of OnLine Dynamic Server. Figure 3-2 shows that OnLine Dynamic Server is in quiescent mode.

```
INFORMIX-OnLine  Version  x.xx.xxx  --  Quiescent  --  Up  xx:xx:xx  --  xxxx Kbytes
```

```
OnLine Dynamic Server is in quiescent mode.
```

Figure 3-2

*Example of onstat
Status Line*

Make a Final Backup

Use **ontape** or ON-Archive to make a level-0 backup and logical-log backup of OnLine Dynamic Server. If you use **ontape**, execute the following command to make a level-0 backup:

```
ontape -s
```

The tape parameters must specify a valid tape device. Be sure to retain and properly label the tape volume that contains the backup. (For more information about making backups, refer to the *INFORMIX-OnLine Dynamic Server Archive and Backup Guide*.)

If you use ON-Archive, execute the following command to make a full-system, level-0 backup:

```
Onarchive> ARCHIVE/DBSPACESET=*
```

Once you have made a level-0 backup, you should also perform a complete backup of the logical log, including the current logical-log file.

Bring OnLine Dynamic Server Off-Line

Execute the following command to shut down OnLine Dynamic Server, Version 7.2:

```
onmode -ky
```

Choose the **Status** option of ON-Monitor to verify that OnLine Dynamic Server is in off-line mode. (For more information about how to use ON-Monitor, refer to the *INFORMIX-OnLine Dynamic Server Administrator's Guide*.) Execute the following command to start ON-Monitor:

```
onmonitor
```



Tip: Use **onmonitor** instead of **onstat** to verify the operating mode. The **onstat** utility is not designed to return the operating-mode status when OnLine Dynamic Server is off-line.

The third line of the ON-Monitor main menu contains the status of OnLine Dynamic Server. The screen in Figure 3-3 indicates that OnLine Dynamic Server is off-line.

```
INFORMIX-OnLine:  Status      Parameters    Dbspaces    Mode    Force-Ckpt ...
Status menu to view INFORMIX-OnLine
-----Off-Line-----Press CTRL-W for Help -----
```

OnLine Dynamic Server is off-line.

Figure 3-3
ON-Monitor
Main Menu

OnLine Dynamic Server must be off-line because Universal Server uses the same files. You cannot install Universal Server if any of the files that it uses are active. Bring OnLine Dynamic Server off-line to ensure that all common files are inactive.

Important: Make a final backup for each OnLine Dynamic Server instance that you plan to convert.



Reconfigure the UNIX Operating System

You might need to change some of the kernel parameters for your UNIX operating system before you install Universal Server. To reconfigure the operating system, follow the directions in the machine-notes file included on your Universal Server distribution media and the kernel-configuration instructions for your operating system. For information on the location of the machine-notes file, refer to “[Documentation Notes, Release Notes, Machine Notes](#)” on page 16 of the Introduction.

Install Universal Server

You must be user **root** to install Universal Server. Set the **SINFORMIXDIR** environment variable to the directory where you plan to install Universal Server.



Warning: *If you install Universal Server in the same directory where OnLine Dynamic Server resided, the installation script overwrites the older files. If you wish to preserve your OnLine Dynamic Server files, you must install Universal Server in a different directory.*

Before you overwrite OnLine Dynamic Server, you must take the following precautions:

- If you do not have the original media for OnLine Dynamic Server, back up the **SINFORMIXDIR** directory before you install Universal Server.
- Copy the configuration file(s) in **SINFORMIXDIR/etc** to another location on the file system.

Follow the directions in the [INFORMIX-Universal Server Installation Guide](#) to install Universal Server. The installation script installs Universal Server into the **SINFORMIXDIR** directory specified for user **root**. The installation script does not bring Universal Server on-line.

Verify That Environment Variables Are Set Correctly

After you install Universal Server, ensure that the following environment variables, which are required, are set to the correct values:

- INFORMIXSERVER
- ONCONFIG
- PATH
- INFORMIXSQLHOSTS (if used)

Important: The client application looks for the *sqlhosts* file in the *\$INFORMIXDIR/etc* directory. However, you can use the *INFORMIXSQLHOSTS* environment variable to change the location or name of the *sqlhosts* file.

Update the ONCONFIG Configuration Files

You can customize your ONCONFIG configuration file and environment variables to take advantage of the new features introduced by Universal Server. After you observe the performance of Universal Server, you might want to make further adjustments.

For information on configuring Universal Server, refer to the [INFORMIX-Universal Server Administrator's Guide](#). For information about environment variables, refer to the [Informix Guide to SQL: Reference](#). For information about tuning the configuration parameters, refer to the [INFORMIX-Universal Server Performance Guide](#).

Important: Use the same values for *ROOTOFFSET*, *ROOTSIZE*, and *ROOTPATH* that you used for OnLine Dynamic Server.



Add a Communications Support Module (Optional)

You can use either the default authentication policy or *Communications Support Modules* (CSM) with Universal Server. After you install the CSM components, configure the CSM by creating entries in the **concsbm.cfg** file and in the options field of the **sqlhosts** file. For information on how to set up CSM, refer to the [INFORMIX-Universal Server Administrator's Guide](#).

Using Communications Support Modules with Client Applications

Existing client applications do not need to be recompiled or relinked if Universal Server does not use Communications Support Modules. If Universal Server uses a Communications Support Module, client applications must relink with new Informix libraries. The clients must also have CSM installed and configured.

Update the ON-Archive Configuration Files

If you use ON-Archive for your OnLine Dynamic Server backup and restore tool, and you will continue to use it with Universal Server, you might need to update ON-Archive configuration parameters.

During the installation procedure for ON-Archive, the install script checks the **\$INFORMIXDIR/etc** directory for files named **config.arc** and **oper_deflt.arc**. If these files do not exist, the install script provides them. If the files do exist, the install script does not overwrite the files. Instead, the install script provides additional files named **Config.arc** and **Oper_deflt.arc** (note the initial uppercase letters). Compare your current versions (**config.arc** and **oper_deflt.arc**) with the new versions and determine whether new or changed configuration parameters or qualifiers exist. ♦

Bring Universal Server On-Line

Execute the following command to bring Universal Server on-line for the first time:

```
oninit
```

As Universal Server comes on-line for the first time, it modifies certain disk structures. This operation should extend the initialization process by only a minute or two. In the unlikely event that your disks cannot accommodate the growth in disk structures, you will find a message in the message-log file instructing you to run **oncheck** on a table. **Oncheck** will tell you that you need to rebuild an index. You should rebuild the index as instructed.



Warning: Universal Server writes to the logical log with the transactions that result from creating the **sysmaster** database. If you run out of log space before the creation of the **sysmaster** database is complete, Universal Server halts and indicates that you must back up the logical log. Once you have backed up the logical log, Universal Server can finish building the **sysmaster** database.

Check your Universal Server message log for status messages that pertain to bringing Universal Server on-line. For information about any messages in the message log, refer to the [INFORMIX-Universal Server Administrator's Guide](#).



Important: If problems are noted in the message file, solve the problems before you continue to the next step.

Databases under Universal Server contain forty-five system catalog tables. OnLine Dynamic Server, Version 7.2 databases contained twenty-nine. The combined size of the system catalog table has grown about 2000 kilobytes per database. You may need to account for this if you have many databases and limited disk space.

Verify the Integrity of the Data

After Universal Server finishes converting the system catalogs, open each database with DB-Access, and use **oncheck** to verify that no data was corrupted in the migration process. You can verify the integrity of the reserve pages, extents system catalog tables, data, and indexes, as shown in Figure 3-4.

Action	oncheck Command
Check reserve pages	oncheck -cr
Check extents	oncheck -ce
Check system catalog tables	oncheck -cc <i>database_name</i>
Check data	oncheck -cD <i>database_name</i>
Check indexes	oncheck -cI <i>database_name</i>

Figure 3-4
*Commands for
Verifying the Data
Integrity*

Make an Initial Universal Server Backup

Use your Universal Server backup and restore tool (ON-Archive, ON-Bar, or **ontape**) to make a level-0 backup. Do not overwrite the tapes you used earlier when you made your final backup of OnLine Dynamic Server. If you use ON-Archive or **ontape**, refer to the [INFORMIX-Universal Server Archive and Backup Guide](#). If you use ON-Bar, refer to the [INFORMIX-Universal Server Backup and Restore Guide](#).

Important: Do not restore the backed up logical-log files from OnLine Dynamic Server for Universal Server.



Migration Complete

When you finish the level-0 backup, the migration process is complete and users can use Universal Server to access data safely.

Once you successfully migrate to Universal Server, you might want to seek ways to obtain maximum performance. If you created sample queries for comparison, you can use them to characterize the performance differences between OnLine Dynamic Server, Version 7.2, and Universal Server. The results of these comparisons might suggest adjustments to configuration parameters, or to the layout of databases, tables, and chunks. For details on performance topics, refer to the [INFORMIX-Universal Server Performance Guide](#).

Install and Configure DataBlade Modules

After you have successfully migrated to Universal Server, install and register any DataBlade modules, supplied by Informix or third-party vendors, that you want to add to Universal Server. *Registration* is the process that makes the DataBlade code available to use in a particular database. For more information on how to use DataBlade modules, refer to the [Datablade Developer Kit User's Guide](#) and the [BladeManager User's Guide](#).



Important: *Completely test Universal Server with traditional relational data before you start to use DataBlade modules. After you successfully use DataBlade modules, then you can begin to use extended data types, routines, and access methods.*

Reverting Universal Server to OnLine Dynamic Server 7.2

Before you can revert to Version 7.2, remove all Universal Server objects from the databases except those created by the boot scripts in the system catalogs. Do not drop the objects created by the **boot90.sql** and **boot901.sql** scripts because the reversion utility uses them.

This section describes the steps for reverting from Universal Server to OnLine Dynamic Server 7.2 on UNIX. Complete the following steps, described in the next sections:

1. Review the database schema to determine whether reversion is possible.
2. Save copies of the current configuration files.
3. Verify the integrity of the data.
4. Back up the Universal Server data.
5. Remove Universal Server features.
6. Run the reversion utility (**onmode -b**).
7. Modify configuration parameters.
8. Reset environment variables.
9. Remove Communications Support Module settings.
10. Start the desired version of OnLine Dynamic Server.
11. Verify the integrity of the data.
12. Back up the OnLine Dynamic Server data.
13. Return OnLine Dynamic Server to on-line mode.

Determine Whether Reversion is Possible

The easiest reversion scenario is a Universal Server database that does not contain any new features. All you need to do is to run the reversion utility and modify the values of the configuration parameters.

Review the database schema to determine whether reversion to OnLine Dynamic Server is possible. Does the schema file contain SQL statements that OnLine Dynamic Server does not support? Does the database contain features that OnLine Dynamic Server does not support, such as DataBlades, smart large objects, user-defined data types, and user-defined routines? Have any new SPL routines been created in Universal Server?

To review the database schema, execute the **dbschema** utility. The **dbschema** syntax is described in [Chapter 10, “Utilities for Data Migration.”](#) The following example displays complete information about the database, **db1**:

```
dbschema -d db1 -ss
```



Important: *You can revert from Universal Server to OnLine Dynamic Server only if you carefully remove all Universal Server features from the databases. If the databases contain Universal Server features, reversion to OnLine Dynamic Server will fail.*

Save Copies of the Configuration Files

Save copies of the ONCONFIG and **concsn.cfg** files in case you decide to upgrade to Universal Server again. (Only Universal Server uses the **concsn.cfg** file that is used for configuring communications support modules.)

Verify the Integrity of the Data

Execute the following commands to check the integrity of the data:

```
oncheck -cI database_name
oncheck -cD database_name
oncheck -cr
oncheck -cc database_name
```

Back Up the Universal Server Data

Before you begin the reversion, make a complete backup. If you use **ontape** or ON-Archive, refer to the [INFORMIX-Universal Server Archive and Backup Guide](#). If you use ON-Bar, refer to the [INFORMIX-Universal Server Backup and Restore Guide](#).

Remove Universal Server Features

Before you revert, you must remove Universal Server features that OnLine Dynamic Server does not support:

- DataBlade modules
- User-defined routines and user-defined functions
- Indexes over 255 bytes
- Extended data types
- Smart large objects
- Universal Server-specific features
- Stored procedures created under Universal Server
- Secondary access methods
- Virtual tables in extspaces
- Sbspaces and extspaces

Uninstall DataBlade Modules

Use BladeManager to unregister all DataBlade modules. Unregistering DataBlade modules removes all data types and routines defined by the DataBlade modules. In UNIX, BladeManager is a command-line utility (**blademgr**) stored in the **\$INFORMIXDIR/bin** directory. For more information, refer to the [BladeManager User's Guide](#).

However, a few hidden tables and error messages remain in each database that BladeManager connects to. These tables contain the list of DataBlades and DataBlade interfaces installed in the database. The hidden table names are: **sysbldregistered**, **sysbldirequired**, **sysbldipprovided**, **sysbldobjects**, and **sysbldobjdepends**. The error messages are in the **syserrors** system catalog and have the **sqlstate** field beginning with **UGENx**.

After you unregister the DataBlades, execute the following SQL script to delete the hidden tables and error messages from each database:

```
DROP TABLE sysbldregistered; --All DataBlades registered
DROP TABLE sysbldirequired; --Inter-DataBlade dependencies
DROP TABLE sysbldipprovided; --DataBlade interfaces
DROP TABLE sysbldobjects; --All objects created by DataBlade
DROP TABLE sysbldobjdepends; --Dependencies between objects
DELETE FROM syserrors WHERE sqlstate LIKE 'UGEN_';
```

Remove User-Defined Routines and User-Defined Functions

Remove all user-defined routines and functions created in Universal Server because OnLine Dynamic Server does not support them. For information on user-defined routines, refer to [Extending INFORMIX-Universal Server: User-Defined Routines](#).

Drop Indexes Over 255 Bytes

In Version 7.2, the maximum key length of an index is 255 bytes. For Universal Server, the maximum key length of an index is 390 bytes. If your indexes have keys longer than 255 bytes, you must drop them before reverting the database.

Remove Extended Data Types and Smart Large Objects

Universal Server provides many data types that OnLine Dynamic Server does not support. Before you revert, drop tables, columns, views, and indexes containing the following data types:

- built-in data types not available in OnLine Dynamic Server:
 - BOOLEAN
 - INT8
 - LVARCHAR
 - SERIAL8
- smart large objects (CLOB and BLOB)
- user-defined data types (OPAQUE and DISTINCT)
- collection data types (SET, MULTISSET, LIST)
- row data types (ROW)
- data types provided by DataBlade modules

If you do not want to drop the tables, columns, or views, you can change the data to a type that OnLine Dynamic Server supports. For example, if the database contains one new table with user-defined data types, either delete that table, or change the user-defined data types to legacy data types and drop the user-defined data types from the system catalogs before you revert.

For information on Universal Server data types, refer to the [Informix Guide to SQL: Reference](#) and [Extending INFORMIX-Universal Server: Data Types](#).

Remove Universal Server-Specific Features

Before you revert, you must remove Universal Server-specific features from client applications and the databases. For example, OnLine Dynamic Server does not support operators and casts to any data type. For information, refer to the [Informix Guide to SQL: Syntax, Extending INFORMIX-Universal Server: Data Types](#), and the [INFORMIX-ESQL/C Programmer's Manual](#).

Remove Stored Procedures Created in Universal Server

Before you revert, drop all stored procedures (SPLs) created in Universal Server. Both OnLine Dynamic Server and Universal Server support stored procedures. However, the internal structure of a stored procedure created in Universal Server is not backward-compatible and does not run under OnLine Dynamic Server. For information on SPLs, refer to the [Informix Guide to SQL: Syntax](#).

Drop Secondary Access Methods (Indexes)

Drop the following secondary access methods (indexes) that OnLine Dynamic Server does not support:

- Generic B-tree indexes on user-defined and built-in data types
- R-tree indexes on spatial data such as maps and diagrams
- Functional indexes on values returned from user-defined functions
- User-defined indexes that a DataBlade module (such as Excalibur Text DataBlade) provides

For more information on indexes, refer to the [INFORMIX-Universal Server Performance Guide](#).

Remove Virtual Tables in External Spaces

Drop all virtual tables stored in external spaces and remove the access methods used to access the external data. For more information on primary access methods, refer to the *Virtual-Table Interface Programmer's Manual*.

Remove Sbspaces and External Spaces

You should already have deleted columns containing smart large objects (BLOB and CLOB) and all virtual tables. Now delete all sbspaces and extspaces. To delete these spaces, execute the following command where *spacename* is the name of the sbspaces or extspaces:

```
onspaces -d spacename
```

For more information on sbspaces and external spaces, refer to the [*INFORMIX-Universal Server Administrator's Guide*](#).

Run the Reversion Utility

Universal Server must be running when you execute the reversion utility. The reversion utility detects and lists remaining Universal Server-specific features that you should remove before reversion can complete.

Execute the reversion utility to revert the database to Version 7.2 (see Figure 3-5):

```
onmode -b 7.2
```

After the reversion is complete, Universal Server is off-line. The reversion utility drops the Universal Server system catalogs and restores compatibility so that users can access the data with OnLine Dynamic Server. The reversion utility does not revert changes made to the layout of the data that do not affect compatibility.

For more information about the **onmode -b** command, refer to [“The onmode Utility” on page 10-63](#).

Figure 3-5
Reverting to an Earlier OnLine Dynamic Server

Revert from	Revert to	Command
Universal Server	Version 7.2	onmode -b 7.2

Modify Configuration Parameters

Remove the following configuration parameters that OnLine Dynamic Server does not support:

- SBSPACENAME
- VPCLASS

You might also need to adjust the values of existing configuration parameters. Alternatively, you can replace the Universal Server ONCONFIG file with the OnLine Dynamic Server ONCONFIG file that you used before you upgraded.

Reset Environment Variables

Reset the environment variables to values that are appropriate for OnLine Dynamic Server. Also remove the following environment variables that OnLine Dynamic Server does not support:

- INFORMIXCONCSMCFG
- INFORMIXKEYTAB

Remove Communications Support Module Settings

If your Universal Server used Communications Support Modules, remove the **csml** option settings from the **sqlhosts** entries for the database server. Otherwise the older database server will return an invalid **sqlhosts** options error. Also delete the **concsml.cfg** file.

Install and Start OnLine Dynamic Server

Install and configure OnLine Dynamic Server according to the instructions in the *INFORMIX-OnLine Dynamic Server Administrator's Guide*.

Execute the **oninit -s** command to bring OnLine Dynamic Server to quiescent mode.

Verify the Integrity of the Data

Before you allow users to access the databases, check the integrity of the data. Follow the steps described in [“Verify the Integrity of the Data” on page 3-11](#).

Back Up OnLine Dynamic Server Data

After you complete the reversion, make a level-0 backup. Use either ON-Archive or the **ontape** utility to make the backup. For information about ON-Archive and **ontape**, refer to the *INFORMIX-OnLine Dynamic Server Archive and Backup Guide*.

Important: Do not overwrite the tapes that you used to back up Universal Server

Return OnLine Dynamic Server to On-Line Mode

To bring OnLine Dynamic Server on-line, execute the **onmode -m** command. The reversion is now complete, and users can access the converted data.



Migrating OnLine Dynamic Server 6.0 or Later to 7.2x

Changes Introduced by OnLine Dynamic Server	4-6
Environment Variable Changes in Version 7.2x	4-6
Environment Variables Introduced in Version 7.2.	4-6
Environment Variables Maintained for Backward Compatibility	4-7
Configuration Parameter Changes in Version 7.2x	4-8
Configuration Parameters Introduced in Version 7.22	4-8
Configuration Parameter Changes in Version 7.22	4-8
Configuration Parameters Introduced in Version 7.2.	4-8
Configuration Parameters Dropped in Version 7.2x	4-9
Global Language Support Changes in Version 7.2x	4-9
Environment Variable Changes in Version 7.1UD1.	4-9
Configuration Parameter Changes in Version 7.1UD1	4-10
Configuration Parameters Introduced in Version 7.1UD1	4-10
Configuration Parameters Dropped in Version 7.1UD1.	4-10
Configuration Parameters Moved in Version 7.1UD1	4-11
Changes to BlobSpace Requirements in Version 7.1UD1	4-11
Environment Variable Changes in Version 7.1	4-12
Configuration Parameter Changes in Version 7.1	4-12
ON-Archive Changes in Version 7.1.	4-12
Environment Variable Changes in Version 6.0	4-13
Preparing to Migrate Between Versions	4-13
Install the Latest Maintenance Release for the Current Version	4-14
Check Available Space and System Requirements	4-15
Save Copies of the Current Configuration Files	4-15
Close all Transactions.	4-16
Initiate a Fast Recovery	4-17
Verify the Integrity of the Data	4-17
Back Up Your OnLine Dynamic Server Files	4-18
Preparation Complete	4-19

Upgrading to OnLine Dynamic Server Version 7.2x: UNIX	4-19
Bring OnLine Dynamic Server Off-Line	4-20
Reconfigure the UNIX Operating System	4-20
Install OnLine Dynamic Server 7.2x (UNIX)	4-20
Verify That Environment Variables Are Set Correctly	4-21
Update the ONCONFIG Configuration Files	4-21
Update the ON-Archive Configuration Files	4-22
Configure the Database Server to Use Enterprise Replication (Version 7.22 Only)	4-22
Bring OnLine Dynamic Server 7.2x On-Line	4-23
Verify the Integrity of the Data	4-24
Check Space Requirements for Blobs	4-24
Make an Initial OnLine Dynamic Server 7.2x Backup	4-24
Run UPDATE STATISTICS	4-24
Verify the Access Path of Your SQL Statements	4-25
Migration Complete	4-25
Upgrading OnLine Dynamic Server to Version 7.22: Windows NT	4-26
Bring OnLine Dynamic Server Off-Line	4-26
Install the Database Server and the Administration Tools	4-27
Installing and Configuring onsnmp	4-28
Customize the Database Server Environment	4-28
Customizing Configuration Parameters	4-28
Customizing Environment Variables	4-28
Configure the Database Server to Use Workgroup Replication (Version 7.22 Only)	4-29
Bring OnLine Dynamic Server On-Line	4-29
Verify the Integrity of the Data	4-30
Make a Complete Backup of OnLine Dynamic Server.	4-30
Run Update Statistics	4-30
Verify the Access Path of Your SQL Statements	4-31
Migration Complete	4-31
Changing Database Server Definitions	4-32
Migrating to a GLS Locale	4-32
Reverting to OnLine Dynamic Server: UNIX	4-33
Remove Unsupported SQL Features	4-34
Remove Fragmented Tables (Version 6.0 Only)	4-34
Save Catalog Information	4-34
Stop Enterprise Replication (Version 7.22 Only).	4-35
Run the Reversion Utility	4-35

Remove GLS Features	4-36
Modify Configuration Parameters	4-37
Configuration Changes for Version 7.22.	4-37
Configuration Changes for Version 7.1UD1 and 7.1x	4-37
Configuration Changes for Version 7.1	4-38
Configuration Changes for Version 6.0	4-39
Reset Environment Variables.	4-40
Environment Variable Changes for Version 7.1UD1 through 7.1x	4-40
Environment Variable Changes for Version 7.1	4-40
Environment Variable Changes for Version 6.0	4-41
Modify the sqlhosts File	4-41
Rename ON-Archive Files (Version 6.0 Only)	4-42
Start the Desired Version of OnLine Dynamic Server	4-42
Restore Catalog Information (Version 6.0 Only)	4-42
Verify the Integrity of the Data	4-42
Back Up OnLine Dynamic Server Data	4-43
Return OnLine Dynamic Server to On-Line Mode	4-43
Reverting from OnLine Dynamic Server 7.1x to 7.1	4-43
Reverting OnLine Dynamic Server: Windows NT	4-44
Remove Unsupported SQL Features	4-44
Uninstall Workgroup Replication Manager (Version 7.22 Only)	4-45
Run the Reversion Utility	4-45
Uninstall OnLine Dynamic Server and Administration Tools.	4-46
Remove GLS Features	4-46
Modify Configuration Parameters	4-47
Configuration Changes for Version 7.12.	4-47
Reset Environment Variables.	4-47
Environment Variable Changes for Version 7.12	4-47
Reinstall the Earlier Version of OnLine Dynamic Server	4-48
Bring OnLine Dynamic Server On-Line	4-49
Verify the Integrity of the Data	4-50
Back Up the OnLine Dynamic Server Data	4-50
Reversion is Now Complete	4-50

T

his chapter describes the procedures for migrating between INFORMIX-OnLine Dynamic Server, Version 7.2x, and earlier versions. The term “*Version 7.2x*” refers to versions 7.2, 7.21, or 7.22. For example, you can use the same procedure for upgrading to Version 7.2 or Version 7.22 on UNIX.

When you upgrade to OnLine Dynamic Server 7.2x, you might want to change database applications to take advantage of new features that OnLine Dynamic Server 7.2x offers. The release-notes files that accompany the software distribution describe these features.

This chapter covers the following topics:

- Changes introduced by OnLine Dynamic Server
- Preparing to migrate
- Upgrading OnLine Dynamic Server to Version 7.2x and installing the administration tools
- Reverting OnLine Dynamic Server to an earlier version

You can upgrade from or revert to the following versions on UNIX:

- OnLine Dynamic Server 6.0
- OnLine Dynamic Server 7.1
- OnLine Dynamic Server 7.1UD1 through 7.1x

You can upgrade from or revert to OnLine Dynamic Server 7.12 on Windows NT.



Tip: For information on upgrading from pre-6.0 versions of OnLine, refer to [Chapter 5, “Migrating Pre-6.0 to 7.2x OnLine Dynamic Server: UNIX.”](#) For information on moving OnLine Dynamic Server data between UNIX and Windows NT environments, refer to [Chapter 8, “Moving Between Database Servers.”](#)

Changes Introduced by OnLine Dynamic Server

This section describes the changes introduced by OnLine Dynamic Server that affect either migration or initial configuration.

Environment Variable Changes in Version 7.2x

OnLine Dynamic Server, Version 7.2x, introduces several new environment variables and maintains several otherwise obsolete environment variables for backward compatibility.

Environment Variables Introduced in Version 7.2

Figure 4-1 shows the new environment variables for Version 7.2x. Review the descriptions of these environment variables to determine whether you need to set them. The **Reference** column in Figure 4-1 indicates the manuals that provide information about these environment variables. Figure 4-1 uses the following abbreviations for Informix manuals:

- HPL: *Guide to the High-Performance Loader* (available on UNIX only)
- GLS: *Guide to GLS Functionality*
- ESQ/C: *INFORMIX-ESQ/C Programmer's Manual*
- REF: *Informix Guide to SQL: Reference*

Figure 4-1
Environment Variables Introduced in Version 7.2

Environment Variable	Variable Affects	Reference
CC8BITLEVEL	ESQ/C only	GLS
CLIENT_LOCALE	Client applications only	GLS
DBCENTURY	SQL APIs only	REF, GLS
DBONPLOAD	High-Performance Loader only	REF, HPL
DB_LOCALE	Database locale	GLS
ESQLMF	ESQ/C compilation	GLS

(1 of 2)

Environment Variable	Variable Affects	Reference
GLS8BITSYS	8-bit clean	GLS
GL_DATE	Date format	GLS
GL_DATETIME	Time format	GLS
NODEFDAC	Default privileges	REF
ONPLOAD	High-Performance Loader	REF, HPL
PLCONFIG	High-Performance Loader	REF, HPL
SERVER_LOCALE	Database server locale	GLS
THREADLIB	ESQL/C only	REF, ESQL/C

(2 of 2)

NLS

Environment Variables Maintained for Backward Compatibility

Version 7.2x supports the environment variables in the following list for backward compatibility with earlier Informix products. If you do not have databases and applications from pre-7.2 versions, you would not use these environment variables. These environment variables are described in Version 7.1 of the *Informix Guide to SQL: Reference*. The interaction of these environment variables with variables that control GLS is described in the *Guide to GLS Functionality*. The following environment variables maintain backward compatibility:

- COLLCSHAR
- DBAPICODE
- DBNLS
- LANG
- LC_COLLATE
- LC_CTYPE
- LC_MONETARY
- LC_NUMERIC
- LC_TIME ♦

Configuration Parameter Changes in Version 7.2x

OnLine Dynamic Server, Version 7.2x, includes new configuration parameters that might affect your installation. You might also need to adjust the values of existing parameters. These configuration parameters are described in the *INFORMIX-OnLine Dynamic Server Administrator's Guide*.

Configuration Parameters Introduced in Version 7.22

Enterprise Replication and Workgroup Replication, introduced in Version 7.22, use the following new configuration parameters:

- CDR_LOGBUFFERS
- CDR_EVALTHREADS
- CDR_DSLOCKWAIT
- CDR_QUEUEMEM
- CDR_NIFUSEHELP
- CDR_NIFMEMS
- CDR_NIFQUEUES

For more information on these configuration parameters, see *Guide to Informix Enterprise Replication* or *Guide to Informix Workgroup Replication*.

Configuration Parameter Changes in Version 7.22

The default value of ALARMPROGRAM has changed.

Configuration Parameters Introduced in Version 7.2

Version 7.2x uses the following new configuration parameters:

- BUFFERS (new definition)
- HETERO_COMMIT
- MAX_PDQPRIORITY (new definition)

Configuration Parameters Dropped in Version 7.2x

Version 7.2x does not use the following OnLine Dynamic Server configuration parameters:

- **BUFSIZE**
- **PDQPRIORITY** (the default value is always zero)

If you do not set PDQPRIORITY through the environment variable or SQL statement, PDQ is turned off for queries (PDQPRIORITY = 0).

Global Language Support Changes in Version 7.2x

NLS

OnLine Dynamic Server, Version 6.0, introduced Native Language Support (NLS). NLS supports single-byte locales, but not multibyte locales. OnLine Dynamic Server for Windows NT, Version 7.12, supports NLS. ♦

GLS

Informix Version 7.2x products use Global Language Support (GLS). GLS lets OnLine Dynamic Server, Version 7.2x, handle different languages, cultural conventions, and code sets for Asian, European, Latin American, and Middle Eastern countries. a discusses the migration implications of GLS. The *Guide to GLS Functionality* provides a full description of GLS. ♦

Environment Variable Changes in Version 7.1UD1

OnLine Dynamic Server, Version 7.1UD1, introduced the following new environment variables:

- **INFORMIXOPCACHE**
- **INFORMIXSQLHOSTS**
- **NODEFDAC**
- **OPTCOMPIND** (new definition)
- **PSORT_NPROCS** (new definition)

Configuration Parameter Changes in Version 7.1UD1

OnLine Dynamic Server, Version 7.1UD1, introduced several new configuration parameters and dropped several others.

Configuration Parameters Introduced in Version 7.1UD1

OnLine Dynamic Server, Version 7.1UD1, introduced the following configuration parameters:

- LBUPRESERVE
- ONDBSPACEDOWN
- OPCACHEMAX
- OPTCOMPIND (new default value)

Configuration Parameters Dropped in Version 7.1UD1

OnLine Dynamic Server, Version 7.1UD1, dropped the following configuration parameters. OnLine Dynamic Server allocates resources dynamically for the structures that these parameters controlled in previous releases:

- CHUNKS
- DBSPACES
- USERTHREADS
- TBLSPACES
- TRANSACTIONS

You might need to reset the value of the LOCKS configuration parameter because it previously depended on the value of TRANSACTIONS.

Configuration Parameters Moved in Version 7.1UD1

OnLine Dynamic Server, Version 7.1UD1, moved the following configuration parameters from the ONCONFIG file (\$INFORMIXDIR/etc/\$ONCONFIG) into the audit configuration file (\$INFORMIXDIR/aaodir/adtcfg.std):

- ADTPATH (UNIX only)
- ADTSIZE (UNIX only)
- ADTERR (all platforms)
- ADTMODE (all platforms)

NT

In the Windows NT environment, the audit configuration parameters are in the audit configuration file (%INFORMIXDIR%\aaodir\adtcfg). ♦

Changes to BlobSpace Requirements in Version 7.1UD1

Versions of OnLine Dynamic Server before 7.1UD1 marked a blob space page as full if the page was more than one-third full. Version 7.1UD1 uses a threshold of one-half of the page size.

In cases where partition blobs have a random size, both schemes use about the same amount of disk space. However, in certain situations the required disk space changes. If you have many partition blobs that are just larger than one-third of a page but less than one-half of a page, the new scheme reduces your space requirements by a factor of two. On the other hand, if you have partition blobs that are just less than one-third of a page and others that are just less than two-thirds of a page, you might see an increase in disk requirements of about 33 percent.

Environment Variable Changes in Version 7.1

OnLine Dynamic Server, Version 7.1, introduced the following environment variables:

- **DELIMIDENT**
- **FET_BUF_SIZE**

Configuration Parameter Changes in Version 7.1

OnLine Dynamic Server, Version 7.1, introduced the following configuration parameters:

- **ALARMPROGRAM**
- **DATASKIP**
- **DS_MAX_QUERIES**
- **DS_MAX_SCANS**
- **DS_TOTAL_MEMORY**
- **MAX_PDQPRIORITY**
- **OPTCOMPIND**

In addition, the PDQPRIORITY configuration parameter was introduced in Version 7.1, but dropped in Version 7.2x.

In Version 7.1, the default values for the configuration parameters LTXHWM and LTXEHWL changed from 80 and 90 to 50 and 60, respectively. OnLine Dynamic Server initialization provides a warning if your ONCONFIG file contains values for these parameters greater than 50 and 60.

ON-Archive Changes in Version 7.1

The names of the ON-Archive error message and help files changed between Version 6.0 and Version 7.1. If you use a **config.arc** file from Version 6.0, you must change the filenames in **\$INFORMIXDIR/etc/config.arc**. If you use the default **Config.arc** file (note the initial uppercase letter) that is installed with Version 7.2x, you do not need to make any changes. For more information, refer to the *INFORMIX-OnLine Dynamic Server Archive and Backup Guide*.

Environment Variable Changes in Version 6.0

Version 6.0 of OnLine Dynamic Server introduced name changes for environment variables and utilities. Environment variable names that began with **TB** in earlier versions begin with **ON** in OnLine Dynamic Server 6.0 or later. For instance, the **TBCONFIG** environment variable was replaced by the **ONCONFIG** environment variable.

Utility names that began with **tb** in earlier versions begin with **on** in OnLine Dynamic Server 6.0 and later. For instance, **tbcheck** was replaced by **oncheck**. For a complete list of utilities, refer to the *INFORMIX-OnLine Dynamic Server Administrator's Guide*.

If you have not already changed the names of these environment variables and utilities in your shell scripts and in the login files of your users, make the changes now.

Preparing to Migrate Between Versions

When you migrate from one version of OnLine Dynamic Server to another, Informix suggests that you follow these guidelines:

- Review the release notes for all versions of OnLine Dynamic Server for information about new features, installation, and fixes to problems. Modify applications as needed.

The release notes are in the `$INFORMIXDIR/release/en_us/0333` directory on UNIX. ♦

The release notes are in the `%INFORMIXDIR%\release` directory on Windows NT. To access the release notes, click the **Release Notes** icon in the **Informix Administration Tools** program group. ♦

- Check the documentation notes for information about features not covered in the manuals.
- Retain both versions of the Informix product software on disk (if you have enough disk space).
- Retain the installation tapes from both versions of the Informix product software.

UNIX

NT

- Make a level-0 backup of the database server before and after migration.
- Use a test instance of the database server to test the installation and migration procedures. Before you attempt to convert the production database, use a test instance in the desired communication mode to practice bringing the new database server on-line.

Before you upgrade, migrate, or revert OnLine Dynamic Server, complete the following steps, which are described in the next sections:

1. Install the latest maintenance release for the current version.
2. Check available space and system requirements.
3. Save copies of the current configuration files.
4. Close all transactions.
5. Initiate a fast recovery.
6. Verify the integrity of the data.
7. Back up your OnLine Dynamic Server files.

Important: Repeat steps 3 through 7 for each instance of OnLine Dynamic Server that you are migrating.



Install the Latest Maintenance Release for the Current Version

Informix recommends that you install the latest maintenance release for your current database server version before you migrate to a new version, especially if you use Version 5.x or 6.x. In this scenario, you plan to migrate from INFORMIX-OnLine 5.03 to OnLine Dynamic Server 7.22 on UNIX. First, install the latest maintenance release for INFORMIX-OnLine, which is 5.07 (as of February 1997), then migrate to Version 7.22. Many minor changes to the 5.x versions are also in the 7.x versions.

For additional information, refer to the installation guide for your database server and the chapters on installation and configuration in the *INFORMIX-OnLine Dynamic Server Administrator's Guide*.

Check Available Space and System Requirements

OnLine Dynamic Server requires 1100 free pages of logical-log space (around 2000 kilobytes) to build the **sysmaster** database.

NT

Before you install OnLine Dynamic Server and the administration tools, verify that your system meets the minimum space and hardware requirements. OnLine Dynamic Server 7.22 requires 15 to 20 percent more space than the previous version because it includes the administration tools. The administration tools consist of INFORMIX-Command Center and Relational Object Manager. OnLine Dynamic Server runs on Windows NT 3.51 and Windows NT 4.0 on an NTFS drive. The administration tools run on Windows NT 3.51, Windows NT 4.0, and Windows 95 on either a FAT or NTFS drive.

For information on the system requirements, refer to the *INFORMIX-OnLine Dynamic Server Administration Tools and Database Server Installation Guide* and the **read_ods.txt** file in Answers OnLine. ♦

Save Copies of the Current Configuration Files

Save copies of the configuration files for each instance of OnLine Dynamic Server, and the **sqlhosts** information:

UNIX

- **\$INFORMIXDIR/etc/SONCONFIG**
- **\$INFORMIXDIR/etc/onconfig**
- **\$INFORMIXDIR/etc/onconfig.std**
- **\$INFORMIXDIR\$/aaodir/adtcfg.***
- **\$INFORMIXDIR\$/dbssodir/adtmasks.***
- **\$INFORMIXDIR/etc/sqlhosts**
- **\$INFORMIXDIR/etc/tctermcap**
- **\$INFORMIXDIR/etc/termcap**

NT

If you use ON-Archive to back up and restore your OnLine Dynamic Server databases and the logical log, you must also copy and save the following files:

- \$INFORMIXDIR/etc/\$ARC_CONFIG
- \$INFORMIXDIR/etc/config.arc
- \$INFORMIXDIR/etc/oper_deflt.arc ♦

Windows NT users

- %ONCONFIG%
- %INFORMIXDIR%\aaodir\adtcfg.*
- %INFORMIXDIR%\dbssodir\adtmask.* ♦

Close all Transactions

Communicate to client users how long you expect the database server to be off-line for the migration. Terminate all database server processes, and place OnLine Dynamic Server in quiescent mode (also called *administration mode*).

To shut down the system gracefully

1. Execute the **onmode -sy** command.
2. Warn all users that you plan to shut down the database server and wait for them to exit.
3. Execute the **onmode -ky** command.
4. To verify the mode of OnLine Dynamic Server, execute the **onstat -** command.

The first line of the **onstat** output contains the status of OnLine Dynamic Server. Figure 4-2 shows that OnLine Dynamic Server is in quiescent mode.

```
INFORMIX-OnLine  Version  x.xx.xxx  --  Quiescent  --  Up  xx:xx:xx  --  xxxx Kbytes
```

OnLine Dynamic Server is in quiescent mode.

Figure 4-2
Example of *onstat*
Status Line



Initiate a Fast Recovery

Warning: User data can be lost or damaged if you interrupt data transactions. If data transactions are interrupted, shut down and restart the database server in administration mode to initiate a fast recovery.

Execute the following command to enter quiescent mode and initiate a fast recovery of your current database:

```
oninit -s
```

A fast recovery rolls forward all committed transactions and rolls back all incomplete transactions since the last checkpoint and then leaves a new checkpoint record in the log with no open transactions pending. (For more information about fast recovery, refer to the *INFORMIX-OnLine Dynamic Server Administrator's Guide*.)

Verify the Integrity of the Data

Use the **oncheck** utility to verify the integrity of data before you make a level-0 (complete) backup. If you find any problems with the data, fix them before you make the backup. You can verify the integrity of the reserve pages, extents, system catalog tables, data, and indexes. First, you need to obtain a list of the databases on your database server.

To obtain a list of the databases on your database server

1. Execute the following command to start ON-Monitor:


```
onmonitor
```
2. Choose **Status→Databases** from ON-Monitor to get a list of the databases on your database server.

Figure 4-3 lists the commands for verifying data integrity.

Action	oncheck Command
Check reserve pages	oncheck -cr
Check extents	oncheck -ce
Check system catalog tables	oncheck -cc <i>database_name</i>
Check data	oncheck -cD <i>database_name</i>
Check indexes	oncheck -cI <i>database_name</i>

Figure 4-3
*Commands for
Verifying the Data
Integrity*

For information on **oncheck**, refer to the *INFORMIX-OnLine Dynamic Server Administrator's Guide*.

Back Up Your OnLine Dynamic Server Files

Use **ontape** to make a complete (level-0) backup of each OnLine Dynamic Server that you plan to migrate. Execute the following command to make a complete backup:

```
ontape -s
```

NT

The Windows NT environment does not support ON-Archive. ♦

The tape parameters must specify a valid tape device. Be sure to retain and properly label the tape volume that contains the backup. For more information about how to use **ontape** to back up OnLine Dynamic Server, refer to the *INFORMIX-OnLine Dynamic Server Archive and Backup Guide*.

Warning: Backups that you make under older versions of OnLine Dynamic Server are not compatible with Version 7.2x. Do not try to restore these backups to OnLine Dynamic Server, Version 7.2x.



UNIX

Preparation Complete

Now you are ready to upgrade, migrate, or revert OnLine Dynamic Server. For information on upgrading, see [“Upgrading OnLine Dynamic Server to Version 7.22: Windows NT” on page 4-26](#). For information on reverting, see [“Reverting OnLine Dynamic Server: Windows NT” on page 4-44](#).

Upgrading to OnLine Dynamic Server Version 7.2x: UNIX

This section describes how to upgrade OnLine Dynamic Server, Version 6.0 or later to Version 7.2x on UNIX. You should have completed the preparatory steps, described in [“Preparing to Migrate Between Versions” on page 4-13](#).

When you upgrade, you can install and test Version 7.2x with the same database server name, configuration files, environment variables, and **sqlhosts** information that you used for the earlier version. After you install OnLine Dynamic Server, Version 7.2x, and verify that it works, you might want to modify the configuration files and environment variables to take advantage of new features such as Global Language Support (GLS).

When you upgrade OnLine Dynamic Server, complete the following steps, which are described in more detail in the sections that follow:

1. Bring the current OnLine Dynamic Server off-line.
2. Reconfigure the operating system, if necessary.
3. Install the new version of OnLine Dynamic Server.
4. Verify that environment variables are set correctly.
5. Update the ONCONFIG configuration file.
6. Update the backup and restore configuration parameters.
7. Configure the database server for Enterprise Replication. (This step is optional.)
8. Bring OnLine Dynamic Server on-line.
9. Verify the integrity of the data.

10. Make an initial complete backup of OnLine Dynamic Server.
11. Run UPDATE STATISTICS.
12. Verify the access path of your SQL statements.

Bring OnLine Dynamic Server Off-Line

Shut down OnLine Dynamic Server to ensure that all common files are inactive.

OnLine Dynamic Server must be off-line because the older and the newer versions share common files. You cannot install OnLine Dynamic Server if any of the common files are active.

Reconfigure the UNIX Operating System

You might need to change some of the kernel parameters for your UNIX operating system before you install OnLine Dynamic Server, Version 7.2x. To reconfigure the operating system, follow the directions in the machine-notes file included on your OnLine Dynamic Server 7.2x distribution media and the kernel-configuration instructions for your operating system.

Install OnLine Dynamic Server 7.2x (UNIX)

You must be user **root** to install OnLine Dynamic Server 7.2x. Set the **\$INFORMIXDIR** environment variable to the directory where you plan to install OnLine Dynamic Server.

Follow the directions in the *UNIX Products Installation Guide* and the *INFORMIX-OnLine Dynamic Server Administrator's Guide* to install OnLine Dynamic Server 7.2x. The installation script installs OnLine Dynamic Server into the **\$INFORMIXDIR** directory specified for user **root**. The installation script does not bring OnLine Dynamic Server on-line.



Warning: *If you install OnLine Dynamic Server 7.2x in the same directory where the earlier version of OnLine Dynamic Server resides, the newer version overwrites the older files. If you wish to preserve the files for the earlier version, you must install OnLine Dynamic Server 7.2x in a different directory.*

Before you overwrite the older version, you must take the following precautions:

- If you do not have the original media for the older version, back up the **\$INFORMIXDIR** directory before you install OnLine Dynamic Server 7.2x.
- Copy the configuration file(s) in **\$INFORMIXDIR/etc** to another location on the file system.

When you finish the installation and system reconfiguration, exit as user **root** and log in as user **informix**.

Verify That Environment Variables Are Set Correctly

After you install OnLine Dynamic Server 7.2x, ensure that the following required environment variables are set to the correct values:

- **INFORMIXSERVER**
- **ONCONFIG**
- **PATH**
- **INFORMIXSQLHOSTS** (if used)



***Important:** The client application looks for the **sqlhosts** file in the **\$INFORMIXDIR/etc** directory. However, you can use the **INFORMIXSQLHOSTS** environment variable to change the location or name of the **sqlhosts** file.*

For information about environment variables, refer to the *Informix Guide to SQL: Reference*.

Update the ONCONFIG Configuration Files

You can customize your ONCONFIG configuration file and environment variables to take advantage of the new features introduced by Version 7.2x. After you observe the performance of OnLine Dynamic Server, you might want to make further adjustments.

For information on configuring, refer to the *INFORMIX-OnLine Dynamic Server Administrator's Guide*. For information about tuning the configuration parameters, refer to the *INFORMIX-OnLine Dynamic Server Performance Guide*.



Important: Use the same values for *ROOTOFFSET*, *ROOTSIZE*, and *ROOTPATH* that you used in the previous version of OnLine Dynamic Server.

Update the ON-Archive Configuration Files

During the installation procedure for ON-Archive, the install script checks the *SINFORMIXDIR/etc* directory for files named **config.arc** and **oper_deflt.arc**. If these files do not exist, the install script provides them. If the files do exist, the install script does not overwrite the files. Instead, the install script provides additional files named **Config.arc** and **Oper_deflt.arc** (note the initial uppercase letters). Compare your current versions (**config.arc** and **oper_deflt.arc**) with the new versions and determine whether new or changed configuration parameters or qualifiers exist.

Configure the Database Server to Use Enterprise Replication (Version 7.22 Only)

Perform this step only if you plan to use Enterprise Replication with OnLine Dynamic Server for UNIX 7.22. Before you can activate Enterprise Replication, you need to update the *ONCONFIG* and *sqlhosts* files for OnLine Dynamic Server. To activate Enterprise Replication for the first time, follow these steps:

1. Bring down OnLine Dynamic Server.
2. Define one or more **dbserver aliases** for Enterprise Replication.
3. Define a **group name** for Enterprise Replication.
4. Bring OnLine Dynamic Server on-line.
5. Use the Replication Manager GUI to define each database server for replication. This step starts Enterprise Replication on those servers.

For complete information on configuring for Enterprise Replication, refer to the *Guide to Informix Enterprise Replication*.

Bring OnLine Dynamic Server 7.2x On-Line

When you bring OnLine Dynamic Server 7.2x on-line for the first time, bring it first to quiescent mode and then to on-line mode.

Execute the following command to bring OnLine Dynamic Server 7.2x from off-line to quiescent mode:

```
oninit -s
```

If the system is not brought up in quiescent mode, you get the following error when you attempt to initialize the database server, and the database server goes off-line:

```
Open transaction detected when changing log versions.
```



Warning: The logical logs continue to fill with the transactions that result from the creation of the **sysmaster** database. If you run out of log space before the creation of the **sysmaster** database is complete, OnLine Dynamic Server halts with a long-transaction error. Thus, you must back up the logical logs.

Execute the **ontape -a** or **ontape -c** command. Once the logical logs are backed up, processing resumes.

When the system reaches quiescent mode successfully, you can note whether you need to adjust the shared memory and semaphore values for your operating system. Check your OnLine Dynamic Server message log for status messages. For information about semaphores, refer to “[Configuring Semaphore Parameters](#)” on page 5-10.



Important: If you note problems in the message file, solve the problems before you continue to the next step.

Execute the following command to change your OnLine Dynamic Server mode from quiescent mode to on-line mode:

```
onmode -m
```

Verify the Integrity of the Data

Before you allow users to access the databases, use the **oncheck** utility to verify that no data was corrupted in the migration process. You can verify the integrity of the reserve pages, system catalog tables, data, and indexes. For more information, see [“Verify the Integrity of the Data” on page 4-17](#).

Check Space Requirements for Blobs

If you are migrating from OnLine Dynamic Server 6.0 or 7.1, the space required for blobs might have changed. Run **oncheck -cD** on your tables that have blobs. This **oncheck** command modifies the internal bitmaps to show any changes in space availability. It also shows WARNING messages when a bitmap has been changed. For more information, refer to [“Changes to Blobspace Requirements in Version 7.1UD1” on page 4-11](#).

Make an Initial OnLine Dynamic Server 7.2x Backup

Use your OnLine Dynamic Server 7.2x backup tool (ON-Bar, ON-Archive, or **ontape**) to make a level-0 backup. Do not overwrite the tapes you used earlier when you made your final backup of OnLine Dynamic Server. For more information about making a backup, refer to the *INFORMIX-OnLine Dynamic Server Archive and Backup Guide* or the *INFORMIX-OnLine Dynamic Server Backup and Restore Guide*.



Important: Do not restore the backed-up logical logs from the earlier version of OnLine Dynamic Server to the newer version of OnLine Dynamic Server.

Run UPDATE STATISTICS

After you complete the migration procedure, run the UPDATE STATISTICS statement. UPDATE STATISTICS updates the information that OnLine Dynamic Server uses to plan efficient queries. For more information about UPDATE STATISTICS, refer to the *Informix Guide to SQL: Syntax*.

Verify the Access Path of Your SQL Statements

Use the SET EXPLAIN statement to verify that the access path of your SQL statements did not change when you upgraded OnLine Dynamic Server. If you have SET EXPLAIN output from the source database server, run SET EXPLAIN for OnLine Dynamic Server. Compare the SET EXPLAIN output from both the source and target database servers. SET EXPLAIN writes the path that the optimizer chooses for each query to the SET EXPLAIN output file. The optimizer chooses the fastest path of execution for table joins.

The UNIX SET EXPLAIN output filename is **sqexplain.out**.

If the SET EXPLAIN output file shows that a different access path was used, complete the following steps:

1. Check the **OPTCOMPIND** environment variable or configuration parameter.
2. Check the **DBSPACETEMP** environment variable or configuration parameter to ensure that adequate temporary dbspaces are defined. You might need to define more temporary dbspaces.
3. Analyze the query access paths and, if necessary, modify the schema to improve the performance.

Migration Complete

When you finish the level-0 backup, the migration process is complete and users can use OnLine Dynamic Server to access data safely.

Once you successfully migrate to OnLine Dynamic Server 7.2x, you might want to seek ways to obtain maximum performance. If you created sample queries for comparison, you can use them to characterize the performance differences between your earlier version and OnLine Dynamic Server 7.2x. The results of these comparisons might suggest adjustments to configuration parameters, or to the layout of databases, tables, and chunks. For details on topics related to performance, refer to the *INFORMIX-OnLine Dynamic Server Performance Guide*. ♦

NT

Upgrading OnLine Dynamic Server to Version 7.22: Windows NT

This section describes how to upgrade OnLine Dynamic Server, Version 7.12 to Version 7.22 on Windows NT. You should have completed the preparatory steps, described in [“Preparing to Migrate Between Versions” on page 4-13](#).

When you upgrade, you can install and test Version 7.22 with the same database server name, configuration files, and environment variables that you used for the earlier version. After you install OnLine Dynamic Server, Version 7.22, and verify that it works, you might want to modify the configuration files and environment variables to take advantage of new features such as Global Language Support (GLS).

Complete the following migration steps if you are upgrading OnLine Dynamic Server on Windows NT:

1. Bring OnLine Dynamic Server off-line.
2. Install the new version of OnLine Dynamic Server.
3. Customize the database server environment. (This step is optional.)
4. Configure the database server for Workgroup Replication. (This step is optional.)
5. Bring OnLine Dynamic Server on-line.
6. Verify the integrity of the data.
7. Make an initial complete backup of OnLine Dynamic Server.
8. Run UPDATE STATISTICS.
9. Verify the access path of your SQL statements.

Bring OnLine Dynamic Server Off-Line

Shut down OnLine Dynamic Server.

Tip: *The installation program automatically shuts down the old OnLine Dynamic Server and starts the new OnLine Dynamic Server.*



Install the Database Server and the Administration Tools

You need to run the Installation wizard twice. First, upgrade the database server. Then install the administration tools.

The Installation wizard replaces the files but does not reconfigure the database server. If a previous version of the database server is on the computer, the **Upgrade** page appears when you install the new product.

To upgrade the database server

1. When the **Upgrade** page appears, click **Next** for the INFORMIX-OnLine Installation wizard. Click **Next** again to begin the installation procedure.
2. Enter the serial number and serial-number key.
3. The installation program automatically verifies and brings down OnLine Dynamic Server, copies the new files, and preserves the database and dbspace data. Then the installation program starts OnLine Dynamic Server, Version 7.22, with the same configuration and shared-server machine.
4. A dialog box displays the following message:

Your INFORMIX-OnLine Dynamic Server is being launched.

The Installation wizard updates the SQLHOSTS registry keys and OSAHOSTS information automatically.

To install the administration tools

1. In the **Run Installation Again** page, select the **Copy all files, but leave configuration alone** installation option.
2. Enter the serial number and serial-number key.
3. In the **Installation Options** page, select the **Administration Tools** check box.
4. Click **Next** to copy the administration tools files and complete the installation.

After you install the database server, administration tools, or both, the Informix Administration Tools program group is available in your windowing environment. For complete information on installation, refer to the *INFORMIX-OnLine Dynamic Server Administration Tools and Database Server Installation Guide*.

Installing and Configuring onsnmp

Workgroup Replication and the **onsnmp** utility require Windows NT SNMP. The installation and upgrade program checks the registry for the SNMP master agent. If the master agent has not been installed, the program displays a warning message but does not configure the registry for Workgroup Replication or **onsnmp**. If you later choose to install the SNMP master agent, you must run the `%INFORMIXDIR%\bin\inssnmp.exe` command line utility to install the SNMP subagents. You need not reinstall the database server.

Customize the Database Server Environment

The Installation program sets the configuration parameters and environment variables for you. However, you can customize the configuration parameters in the ONCONFIG file and environment variables for OnLine Dynamic Server.

Customizing Configuration Parameters

Use a text editor to edit the ONCONFIG file. (You cannot use ON-Monitor to edit the configuration parameters because that utility is no longer supported.)

The default value of ALARMPROGRAM has changed. For more information, see the *INFORMIX-OnLine Dynamic Server Administrator's Guide*.

Important: Use the same values for *ROOTOFFSET*, *ROOTSIZE*, and *ROOTPATH* that you used for the earlier version of OnLine Dynamic Server.

Customizing Environment Variables

Use the **setnet32** tool to customize the environment variables on the client. For more information on environment variables, refer to the *Informix Guide to SQL: Reference*.



Configure the Database Server to Use Workgroup Replication (Version 7.22 Only)

Perform this step only if you plan to use Workgroup Replication with OnLine Dynamic Server for Windows NT 7.22. Before you can activate Workgroup Replication, you need to update the ONCONFIG file, **services** file, and SQLHOSTS registry for OnLine Dynamic Server.

To activate Workgroup Replication for the first time

1. Bring down OnLine Dynamic Server.
2. Define one or more **dbserver aliases** for Workgroup Replication.
3. Define a **group name** for Workgroup Replication.
4. Bring OnLine Dynamic Server on-line.
5. Use the Replication Manager GUI to define each database server for replication. This step starts Workgroup Replication on those servers
6. Verify that the SNMP service is installed on the host and that the SNMP subagents are registered.

For complete information on configuring for Workgroup Replication, refer to the *Guide to Informix Workgroup Replication*. For information on how to use SNMP, refer to the *Informix SNMP Subagent Guide*.

Bring OnLine Dynamic Server On-Line

The installation program brings OnLine Dynamic Server on-line automatically.

If you customized the database server environment, bring down and restart OnLine Dynamic Server with the Command Center. When you restart OnLine Dynamic Server, the changes to the configuration parameters and environment variables take effect.

To start OnLine Dynamic Server manually using the Command Center

1. In the **Informix Administration Tools** program group, double-click the **Command Center** icon.
2. In the Command Center, select the database server in the **All Servers** tree view or the **Servers** list box.
3. Choose **Server→On-line**.

For more information, refer to *Using the INFORMIX-Command Center* or General help.

Verify the Integrity of the Data

Use the **oncheck** utility to verify the integrity of data, as described in [“Verify the Integrity of the Data” on page 4-17](#).

Make a Complete Backup of OnLine Dynamic Server

Use the Backup and Restore tool to make a complete, whole-system backup of OnLine Dynamic Server. For more information, see “Backing Up and Restoring Data” in *Using the INFORMIX-Command Center* or General help.

Run Update Statistics

After you complete the migration procedure, run the UPDATE STATISTICS statement. The UPDATE STATISTICS statement updates the information that OnLine Dynamic Server uses to plan efficient queries. For more information about UPDATE STATISTICS, refer to the *Informix Guide to SQL: Syntax*.

NT

Verify the Access Path of Your SQL Statements

Use the SET EXPLAIN statement to verify that the access path of your SQL statements did not change when you migrated to OnLine Dynamic Server. If you have SET EXPLAIN output from the source database server, run SET EXPLAIN for OnLine Dynamic Server. Compare the SET EXPLAIN output from both the source and target database servers. SET EXPLAIN writes the access path that the optimizer chooses for each query to the SET EXPLAIN output file. The optimizer chooses the fastest path of execution for table joins.

For Windows NT, the SET EXPLAIN output filename is %INFORMIXDIR%\sqlexpln\<username>.out. ♦

If the SET EXPLAIN output file shows that a different access path was used, complete the following steps:

1. Check the **OPTCOMPIND** environment variable or configuration parameter.
2. Check the **DBSPACETEMP** environment variable or configuration parameter to ensure that adequate temporary dbspaces are defined. You might need to define more temporary dbspaces.
3. Analyze the query access paths, and modify the schema to improve the performance if necessary.

Migration Complete

The first time OnLine Dynamic Server is brought on-line, the **sysmaster** database is built. Check the message log to ensure that the **sysmaster** database build has completed before you allow users to access the database server. After you ensure that client users can access data on OnLine Dynamic Server, the migration process is complete.

Changing Database Server Definitions

The installation program automatically updates the registry information and the database server definitions on the shared-server (SQLHOSTS) machine. The *database server definitions* consist of the database server name, TCP/IP host names, network types, and the service names. Use the **Setup** program if you want to change the **informix** user password, specify a different computer as the shared-server machine, or edit the database server definitions. For example, you might specify a different shared-server machine if you have migrated the database server to a new computer or connected a single client to several database servers.

For information on how to use **Setup** to configure database servers, see *Using the INFORMIX-Command Center* or the General Help. For information on connectivity, see the *INFORMIX-OnLine Dynamic Server Administrator's Guide*.

GLS

Migrating to a GLS Locale

If you wish to migrate OnLine Dynamic Server to a non-English GLS locale, set the DB_LOCALE and CLIENT_LOCALE environment variables before you open the database in Version 7.2x. If your previous database server version used Native Language Support (NLS), replace the NLS environment variables, such as **COLLCHAR**, with GLS environment variables. For information on how to work with locales and how to set GLS environment variables, see the *Guide to GLS Functionality* and [Chapter 9, “Changing Locales.”](#)

Important: This version of GLS does not support Middle Eastern languages. ♦



Reverting to OnLine Dynamic Server: UNIX

This section covers the changes that occur when you revert to an earlier version of OnLine Dynamic Server on UNIX. You can move between the following versions of OnLine Dynamic Server:

- From OnLine Dynamic Server 7.22 to OnLine Dynamic Server 7.2
- From OnLine Dynamic Server 7.2x to OnLine Dynamic Server 7.1UD1 through 7.14
- From OnLine Dynamic Server 7.2x to OnLine Dynamic Server 7.1
- From OnLine Dynamic Server 7.2x to OnLine Dynamic Server 6.0
- From OnLine Dynamic Server 7.2x to OnLine 5.0
- From OnLine Dynamic Server 7.2x to OnLine 4.1
- From OnLine Dynamic Server 7.1x (later version) to OnLine Dynamic Server 7.1x (earlier version)

This section describes the steps for reverting to an earlier version of OnLine Dynamic Server. When you revert, you must consider changes in the definitions of configuration parameters and environment variables. Review the schema file for SQL statements that are not supported in the earlier version of OnLine Dynamic Server.

Follow the preparatory steps, described in [“Preparing to Migrate Between Versions” on page 4-13](#), then complete the following steps:

1. Remove unsupported SQL features.
2. Remove fragmented tables (only when you are reverting to Version 6.0).
3. Save catalog information.
4. Stop Enterprise Replication, if it is installed.
5. Run the reversion utility (**onmode -b**).
6. Remove GLS features (only if GLS was used).
7. Modify configuration parameters.
8. Reset environment variables.
9. Modify the **sqlhosts** file (only for Version 6.0 or Version 7.1).
10. Rename ON-Archive files (only for Version 6.0).

11. Start the desired version of OnLine Dynamic Server.
12. Restore catalog information (only for Version 6.0).
13. Verify the integrity of the data.
14. Back up the OnLine Dynamic Server data.
15. Return OnLine Dynamic Server to on-line mode.

Remove Unsupported SQL Features

Before you revert, you must remove SQL features that the earlier version of OnLine Dynamic Server does not support. See the “New Features of This Product” section in the appropriate version of the *Informix Guide to SQL: Syntax*.

Remove Fragmented Tables (Version 6.0 Only)

If you are reverting to Version 6.0, you must change all fragmented tables back into unfragmented tables. The *INFORMIX-OnLine Dynamic Server Administrator's Guide* includes instructions for defragmenting tables.

Save Catalog Information

If your current OnLine Dynamic Server instance uses secure-auditing masks or ON-Archive, and you wish to preserve the associated catalog information, you must unload these system catalog tables before you continue. Execute the following command to unload the catalog tables:

```
$INFORMIXDIR/etc/smi_unld
```

When the **smi_unld** utility finishes unloading the information, the utility displays instructions for reloading the information. *Save these instructions.*

After you complete the reversion and initialize OnLine Dynamic Server, you can reload the data that you preserved. Follow the instructions given with the **smi_unld** utility for reloading the information. Typically, you should execute the following command:

```
$INFORMIXDIR/etc/smi_load $INFORMIXDIR/etc/
```

Stop Enterprise Replication (Version 7.22 Only)

Skip this section if Enterprise Replication is not installed on your system.

To revert to an earlier version if Enterprise Replication is active

1. Stop Enterprise Replication.
2. For altered tables with CRCOLS, issue the command:

```
alter <table> drop CRCOLS
```
3. Execute the **onmode -b** command, as described in “Run the Reversion Utility” to revert to the earlier version of OnLine Dynamic Server.



Warning: *If you try to revert to a previous version of OnLine Dynamic Server while Enterprise Replication is active, the reversion will fail.*

To revert to an earlier version if Enterprise Replication is inactive

1. In this situation, Enterprise Replication was previously active on this database server. For altered tables with CRCOLS, issue the command:

```
alter <table> drop CRCOLS
```
2. Execute the **onmode -b** command, as described in “Run the Reversion Utility,” to revert to the earlier version of OnLine Dynamic Server. The **syscdr** database is dropped during reversion.

For more information, see the *Guide to Informix Enterprise Replication*.

Run the Reversion Utility

OnLine Dynamic Server must be running when you execute the reversion utility. You must use this reversion utility to restore compatibility before users can access the data with the earlier version. The **onmode** utility does not revert changes made to the layout of the data that do not affect compatibility.

To revert to an earlier version of OnLine Dynamic Server

1. Execute the reversion utility to revert the database, where *version_number* is the earlier OnLine Dynamic Server version (see Figure 4-4):

```
onmode -b version_number
```

2. Drop the **sysutils** database if the earlier version does not support ON-Bar.

The reversion utility forcibly removes all users and shuts down OnLine Dynamic Server. For more information about the **onmode -b** command, refer to [“The onmode Utility” on page 10-63](#).

Figure 4-4
Reverting to an Earlier OnLine Dynamic Server

Revert from	Revert to	Command
Version 7.22	Version 7.2	onmode -b 7.2
Version 7.2 through 7.22	Version 7.1UD1	onmode -b 7.1UD1
Version 7.2 through 7.22	Version 7.1	onmode -b 7.1
Version 7.2 through 7.22	Version 6.0	onmode -b 6.0
Version 7.1UD1 through 7.14	Version 7.1	onmode -b 7.1
Version 7.12 through 7.14	Version 7.1UD1	Reversion automatic



Tip: The **onmode -b** command also rebuilds the user-table indexes automatically.

Remove GLS Features

Skip this step if OnLine Dynamic Server uses the default English locale (**en_us.8859-1**). To revert OnLine Dynamic Server from GLS to Native Language Support (NLS) or Asian Language Support (ALS), set the appropriate NLS or ALS locales and environment variables. For information on working with locales, see the *Guide to GLS Functionality* and [Chapter 9, “Changing Locales.”](#)

Modify Configuration Parameters

Version 7.2x uses configuration parameters that did not appear in earlier versions of OnLine Dynamic Server. In addition, Version 7.2x changed default values of some parameters.

Configuration Changes for Version 7.22

Version 7.22 changed the default value of the ALARMPROGRAM configuration parameter. You might need to revise the values in your configuration file.

Remove the following parameters from your ONCONFIG file:

- CDR_LOGBUFFERS
- CDR_EVALTHREADS
- CDR_DSLOCKWAIT
- CDR_QUEUEMEM
- CDR_NIFUSEHELP
- CDR_NIFMEMS
- SCDR_NIFQUEUES

Configuration Changes for Version 7.1UD1 and 7.1x

If you are reverting to Version 7.1UD1 or 7.1x, remove the following parameters from your ONCONFIG configuration file:

- CHUNKS
- DBSPACES
- HETERO_COMMIT
- TBLSPACES
- TRANSACTIONS
- USERTHREADS

Version 7.2x changed the definition of the following two configuration parameters. You might need to revise the values in your configuration file:

- BUFFERS
- MAX_PDQPRIORITY

Version 7.2x removed the following configuration parameters. If you do not restore the parameters to your ONCONFIG file, OnLine Dynamic Server uses the default values. You can add these parameters to your ONCONFIG file:

- BUFFSIZE
- PDQPRIORITY

Configuration Changes for Version 7.1

If you are reverting to Version 7.1UC1, remove the following parameters from your ONCONFIG configuration file:

- LBUPRESERVE
- ONDBSPACEDOWN
- OPCACHEMAX

Version 7.1UD1 or 7.2 changed the definition of the following configuration parameters. You might need to revise the values in your configuration file:

- BUFFERS (changed in 7.1UD1)
- LOCKS (changed in 7.2)
- MAX_PDQPRIORITY (changed in 7.1UD1)
- OPTCOMPIND (changed in 7.2)

Version 7.1UD1 removed the following configuration parameters. If you do not restore the parameters to your ONCONFIG file, OnLine Dynamic Server uses the default values. You should restore these parameters to your ONCONFIG file:

- CHUNKS
- DBSPACES
- PDQPRIORITY (removed in 7.2)
- TBLSPACES
- TRANSACTIONS
- USERTHREADS

Version 7.1UD1 moved the following audit configuration parameters into the audit configuration file (\$INFORMIXDIR/aaodir/adtcfg.std). If you use the ON-Audit utility, restore these parameters to your ONCONFIG file:

- ADTPATH (UNIX only)
- ADTSIZE (UNIX only)
- ADTERR
- ADTMODE

Configuration Changes for Version 6.0

If you are reverting to Version 6.0, make the changes specified in [“Configuration Changes for Version 7.1UD1 and 7.1x”](#) and [“Configuration Changes for Version 7.1”](#) on page 4-38. In addition, remove the following parameters from your ONCONFIG configuration file:

- ALARMPROGRAM
- DATASKIP
- DS_MAX_QUERIES
- DS_MAX_SCANS
- DS_TOTAL_MEMORY
- MAX_PDQPRIORITY
- OPTCOMPIND
- PDQPRIORITY



Important: Use the same values for *ROOTPATH*, *ROOTSIZE*, and *ROOTOFFSET* in your ONCONFIG configuration file for both versions of OnLine Dynamic Server. *ROOTOFFSET* must equal 0.

Reset Environment Variables

Reset the environment variables to values that are appropriate for your version of OnLine Dynamic Server.

Environment Variable Changes for Version 7.1UD1 through 7.1x

OnLine Dynamic Server, Version 7.1UD1 through 7.1x, supports NLS, not GLS. When you revert, delete the following environment variables:

- CC8BITLEVEL
- CLIENT_LOCALE
- DBCENTURY
- DBONPLOAD
- DB_LOCALE
- ESQLMF
- GLS8BITSYS
- GL_DATE
- GL_DATETIME
- NODEFDAC
- ONPLOAD
- PLCONFIG
- SERVER_LOCALE
- THREADLIB

Environment Variable Changes for Version 7.1

When you revert to Version 7.1, do not use the following environment variables:

- INFORMIXOPCACHE
- INFORMIXSQLHOSTS
- NODEFDAC

The recommended settings of the following environment variables changed between Version 7.1 and Version 7.2x. You might need to reset their values:

- **OPTCOMPIND**
- **PSORT_NPROCS**

Environment Variable Changes for Version 6.0

When you revert to Version 6.0, make the environment variable changes that are specified for Version 7.1. In addition, do not use the following environment variables:

- **DELIMIDENT**
- **FET_BUF_SIZE**
- **PDQPRIORITY**

The **PDQPRIORITY** environment variable and the SQL **SETPDQPRIORITY** statement were introduced after Version 6.0. The **PDQPRIORITY** environment variable does not cause problems if you leave it set for Version 6.0, but it might cause confusion. You must remove the **SETPDQPRIORITY** statement from your Version 6.0 applications.

Modify the sqlhosts File

Version 7.1UD1 introduced the following enhancements to the **sqlhosts** file. If you use any of these enhancements, you must modify your **sqlhosts** file before you run Version 7.1 or Version 6.0:

- Host-name length of 256 characters
- **INFORMIXSQLHOSTS** environment variable
- Stream pipes
- Explicit addressing for TCP/IP
- Options field

Rename ON-Archive Files (Version 6.0 Only)

The names of the ON-Archive error message and help files changed between Version 6.0 and Version 7.1. For the correct names for these files, refer to the *INFORMIX-OnLine Dynamic Server Archive and Backup Guide*, Version 6.0.

Start the Desired Version of OnLine Dynamic Server

To start the earlier version of OnLine Dynamic Server, follow the instructions in the *INFORMIX-OnLine Dynamic Server Administrator's Guide* for that version.

Restore Catalog Information (Version 6.0 Only)

After you initialize OnLine Dynamic Server, you might need to restore catalog information. Follow the instructions in [“Save Catalog Information” on page 4-34](#).

Verify the Integrity of the Data

Before you allow users to access the databases, verify the integrity of the data. Execute the following command to bring OnLine Dynamic Server to quiescent mode:

```
oninit -s
```

Execute the following commands to check the integrity of the data:

```
oncheck -cI database_name
oncheck -cD database_name
oncheck -cr
oncheck -cc database_name
```

Back Up OnLine Dynamic Server Data

After you complete the reversion, Informix recommends that you make a level-0 backup. Use either ON-Archive or the **ontape** utility to make the backup. Do not overwrite the tapes that you used to back up OnLine Dynamic Server 7.2x. For information about making a backup, refer to the *INFORMIX-OnLine Dynamic Server Archive and Backup Guide*.

Return OnLine Dynamic Server to On-Line Mode

To bring OnLine Dynamic Server on-line, execute the following command:

```
onmode -m
```

The reversion is now complete, and users can access the converted data.

Reverting from OnLine Dynamic Server 7.1x to 7.1

OnLine Dynamic Server 7.1 is not compatible with later 7.1x versions of the product. For example, if you wish to revert from OnLine Dynamic Server 7.12 to OnLine Dynamic Server 7.1, you must run the **onmode -b** command to restore the data to a form that is compatible with the earlier version.

OnLine Dynamic Server versions 7.1UD1, 7.11 through 7.14 are compatible with each other, so you do not use **onmode -b** when you migrate the data.

Reverting OnLine Dynamic Server: Windows NT

This section describes the steps for reverting from OnLine Dynamic Server Version 7.22 to Version 7.12 on Windows NT and removing the administration tools.

Follow the preparatory steps, described in [“Preparing to Migrate Between Versions” on page 4-13](#), then complete the following steps:

1. Remove unsupported SQL features.
2. Uninstall Workgroup Replication Manager, if it is installed.
3. Run the reversion utility (**onmode -b**).
4. Uninstall OnLine Dynamic Server and the administration tools.
5. Remove GLS features (only if GLS was used).
6. Modify configuration parameters.
7. Reset environment variables.
8. Reinstall the old version of OnLine Dynamic Server.
9. Start OnLine Dynamic Server without the **-iy** option.
10. Verify the integrity of the data.
11. Back up the OnLine Dynamic Server data.

Remove Unsupported SQL Features

Before you revert, you must remove SQL features that the earlier version of OnLine Dynamic Server does not support. See the “New Features of This Product” section in the appropriate version of the *Informix Guide to SQL: Syntax*.

Uninstall Workgroup Replication Manager (Version 7.22 Only)

Skip this section if Workgroup Replication is not installed on your system.

To revert to an earlier version if Workgroup Replication is active

1. Uninstall Workgroup Replication Manager.
2. Stop Workgroup Replication.
3. Execute the **onmode -b** command, as described in “Run the Reversion Utility,” to revert to the earlier version of OnLine Dynamic Server.



Warning: If you try to revert to a previous version of OnLine Dynamic Server while Workgroup Replication is active, the reversion will fail.

To revert to an earlier version if Workgroup Replication is inactive

In this situation, Workgroup Replication was installed but is not active on this database server.

1. Uninstall Workgroup Replication Manager.
2. Execute the **onmode -b** command, described in “Run the Reversion Utility,” to revert to the earlier version. The **syscdr** database is dropped during reversion.

For more information, see the *Guide to Informix Workgroup Replication*.

Run the Reversion Utility

OnLine Dynamic Server must be running when you execute the reversion utility. Execute the reversion utility to revert the database, where *version_number* is the earlier OnLine Dynamic Server version (see [Figure 4-5 on page 4-46](#)):

```
onmode -b version_number
```

After the reversion is complete, OnLine Dynamic Server is off-line. For more information about the **onmode** command, refer to [“The onmode Utility” on page 10-63](#).

Revert from	Revert to	Command
Version 7.2x	Version 7.12	onmode -b 7.1
Version 7.22	Version 7.2	onmode -b 7.2

Figure 4-5
*Reverting to an Earlier
OnLine Dynamic Server*

Uninstall OnLine Dynamic Server and Administration Tools

Uninstall both the database server and the administration tools. Earlier versions of OnLine Dynamic Server do not support the administration tools.

To uninstall the product

1. Double-click the **Uninstall** icon in the **Informix Administration Tools** program group.
2. In the Uninstall dialog box, check **Remove OnLine Server**.

Warning: Do not check **Remove all OnLine databases, supporting files and all database information**. If you check this option, your configuration, dbspaces, and database information will be lost, making reversion impossible.

3. To uninstall the administration tools, check **Remove Administration Tools**.
4. Click **OK** to uninstall OnLine Dynamic Server.

Remove GLS Features

Skip this step if OnLine Dynamic Server uses the default English locale (**en_us.8859-1**). To revert OnLine Dynamic Server from GLS to Native Language Support (NLS), set the appropriate NLS locales and environment variables. For information on working with locales, see the *Guide to GLS Functionality* and [Chapter 9, “Changing Locales.”](#)



Modify Configuration Parameters

Version 7.22 uses configuration parameters that did not appear in earlier versions of OnLine Dynamic Server. In addition, Version 7.22 changed default values of some parameters. For more information, see [“Configuration Parameters Introduced in Version 7.22” on page 4-8](#).

Configuration Changes for Version 7.12

You must add the ONLANGMAP configuration parameter to your ONCONFIG file before you start OnLine Dynamic Server 7.12. ONLANGMAP takes the value **en_US-English**. For information on ONLANGMAP, see the INFORMIX-OnLine Dynamic Server *for Windows NT* 3.51, Version 7.12.TC2 documentation notes.

If you are reverting to Version 7.12, remove the HETERO_COMMIT configuration parameter from your ONCONFIG file.

Version 7.22 changed the default value of the ALARMPROGRAM configuration parameter. You might need to revise the values in your ONCONFIG file.

Reset Environment Variables

Reset the environment variables to values that are appropriate for your version of OnLine Dynamic Server. For information on environment variables, refer to the *Informix Guide to SQL: Reference*.

Environment Variable Changes for Version 7.12

OnLine Dynamic Server, Version 7.12, supports NLS, not GLS. When you revert to Version 7.12, delete the following environment variables:

- CC8BITLEVEL
- CLIENT_LOCALE
- DBCENTURY
- DBFLTMASK
- DBONPLOAD
- DB_LOCALE
- ESQLMF

- GLS8BITFSYS
- GL_DATES
- GL_DATETIME
- ONPLOAD
- PLCONFIG
- SERVER_LOCALE
- THREADLIB

Add the following environment variables:

- DBNLS
- COLLCHAR
- LANG

Reinstall the Earlier Version of OnLine Dynamic Server

Reinstall the earlier version of OnLine Dynamic Server in the same directory as the Version 7.22 files.

To perform the installation

1. On the **Run Installation Again** page, select the **Copy all files, but leave configuration alone** installation option.
2. Supply your **Serial Number** and **Serial Number Key**, as shown on the serial-number key card.
3. Select one or both of the components you want to install: **OnLine Database Server** or **Administration Tools**. Click **Next**.
4. The installation program automatically copies the 7.12 database server files and saves the configuration and the database data.

For more information on installation, refer to the *INFORMIX-OnLine Dynamic Server Administration Tools and Database Server Installation Guide*.

Warning: Do not select **Copy all files and reconfigure the product**. If you select this option, your configuration and database information will be lost.





Warning: You must add the *ONLANGMAP* parameter to the *ONCONFIG* file before you start OnLine Dynamic Server 7.12 containing the reverted databases. *ONLANGMAP* takes the value **en_US-English**. The *ONLANGMAP* configuration parameter ensures backward compatibility with non-NLS locales on Windows NT. If the database server crashes on install, just restart it.

Bring OnLine Dynamic Server On-Line

Start OnLine Dynamic Server without the **-iy** option. OnLine Dynamic Server initializes the shared memory and builds the **sysmaster** database. After the **sysmaster** database is built, the reversion process is complete.

To start OnLine Dynamic Server

1. From the Windows NT **Main** program group, double-click the **Control Panel** icon.
2. Double-click the **Services** icon.
3. Select **INFORMIX-OnLine Dynamic Server** from the Services list box.
4. Click **Start**.

The Services dialog box displays the status of the database server.



Warning: If you start OnLine Dynamic Server with the **-iy** parameters after the first time, it will overwrite the existing root dbspace unless you have first changed the *ROOTPATH* parameter in the *ONCONFIG* file. If OnLine Dynamic Server overwrites the existing root dbspace, it destroys the information that the root dbspace contains, including information about any databases that you have created. Consequently, you must then restore them from backup tapes.

Verify the Integrity of the Data

To check the integrity of the data, follow the steps described in [“Verify the Integrity of the Data” on page 4-17](#).

Back Up the OnLine Dynamic Server Data

After you complete the reversion, Informix recommends that you make a complete backup. Use the **ontape** utility to make the backup. For information about how to use **ontape**, refer to the *INFORMIX-OnLine Dynamic Server Archive and Backup Guide*.

Important: *Do not overwrite the tapes that you used to back up OnLine Dynamic Server, Version 7.22.*



Reversion is Now Complete

Ensure that client users can access data on the earlier version of OnLine Dynamic Server.

Migrating Pre-6.0 to 7.2x OnLine Dynamic Server: UNIX

Preparation for Migration Between Versions	5-3
Changes Introduced in Version 6.0	5-4
Changes to Database Utilities	5-5
Planning for Migration	5-6
Operating-System Configuration Issues	5-6
Estimating the Size and Number of Shared-Memory Segments	5-6
Configuring Semaphore Parameters	5-10
Configuring the Number of Open-File Descriptors	5-12
Operating-System Updates.	5-12
Disk-Utilization Issues	5-13
Accommodating System Catalog Indexes.	5-14
Accommodating the sysmaster Database	5-15
Locating Temporary Files and Tables	5-15
Accommodating Data Distributions.	5-16
Accommodating User-Table Indexes	5-17
Accommodating the Conversion of User-Table Indexes	5-18
Managing Secure-Auditing Log Files	5-20
Database System-Configuration Issues	5-21
Changing Environment Variables	5-21
Revising Configuration Parameter Values	5-21
Increasing Logical-Log Space	5-22
Planning for Additional Shared-Memory Usage	5-22
Saving Pre-Existing sysmaster Databases	5-23
Configuring Secondary Database Servers for Data Replication	5-23
Managing Backups	5-23
Making Performance Comparisons	5-23

Migration from OnLine 4.1 or 5.0 on UNIX to OnLine Dynamic	
Server 6.0 or Later	5-24
Install the Latest Maintenance Release for the Current Version . . .	5-24
Capture Configuration and Chunk-Layout Information	5-25
Close All Transactions and Make a Final Level-0 Backup	5-26
Bring OnLine 4.1 or 5.0 Off-Line	5-28
Install and Configure OnLine Dynamic Server 7.2x	5-29
Reconfigure the UNIX Operating System	5-31
Bring OnLine Dynamic Server 7.2x On-Line	5-32
Convert User-Table Indexes	5-33
Move Tables to Another Dbspace	5-34
Use the oncheck Utility to Convert Indexes	5-35
Use UPDATE STATISTICS to Convert Indexes	5-36
Use DROP INDEX and CREATE INDEX to Convert Indexes . . .	5-37
Verify the Integrity of the Data	5-37
Make an Initial Backup Under OnLine Dynamic Server 7.2x . . .	5-37
Migration Complete	5-38
Global Language Support	5-38
Reverting to OnLine 5.0	5-39
Save Configuration Files.	5-39
Remove All Users from OnLine Dynamic Server	5-39
Verify the Integrity of the Data	5-40
Back Up OnLine Dynamic Server Data.	5-40
Remove Features Introduced by Later Versions of OnLine	
Dynamic Server	5-40
Execute the Reversion Utility	5-41
Prepare the TBCONFIG Configuration File	5-42
Reset Environment Variables	5-42
Bring Up the Desired Version of OnLine Server	5-42
Verify the Integrity of the Data	5-43
Back Up OnLine 5.0 Server Data	5-43
Return OnLine 5.0 to On-Line Mode	5-43

This chapter describes upgrading from pre-6.0 versions of INFORMIX-OnLine to INFORMIX-OnLine Dynamic Server, Version 7.2x, on UNIX. You can use the same procedure for upgrading to versions 7.2, 7.21, or 7.22.

Preparation for Migration Between Versions

When you migrate from one version of any database server to another, Informix suggests that you follow these guidelines:

- Check the release notes for information about the proper operating-system release and any patches that are required for installation of the database server.
The release notes are in the `$INFORMIXDIR/release/en_us/0333` directory on UNIX.
- Check the documentation notes for information about features not covered in the manuals.
- Retain both versions of the Informix product software on disk (if you have enough disk resources).
- Retain the installation tapes from both versions of the Informix product software.
- Make a level-0 backup of the database server before and after migration.
- Use a test instance of the database server to test the installation and migration procedures. Use a test instance in the desired communication mode to practice bringing the new database server on-line before you attempt to convert the production database.

In addition to the material in this chapter, read the following information in [“Changes Introduced by OnLine Dynamic Server” on page 4-6](#).



Refer also to both the *UNIX Products Installation Guide* and the chapters on installation and configuration in the *INFORMIX-OnLine Dynamic Server Administrator's Guide*.

Tip: This chapter does not discuss migration to OnLine Dynamic Server for Windows NT. For information on migration to OnLine Dynamic Server for Windows NT, see [Chapter 8, “Moving Between Database Servers.”](#)

Changes Introduced in Version 6.0

In Version 6.0, OnLine Dynamic Server introduced an architecture that differs greatly from the architecture used in OnLine 4.1 or 5.0. This new architecture requires significant changes in allocation and disk use. Figure 5-1 describes the major changes.

Figure 5-1
Changes Introduced in Version 6.0

Area of Change	Comments
Backup-tape format	Backups and logical-log backups made with pre-6.0 tb tape are not compatible with either of the two Version 7.2x tape utilities, ON-Archive and ontape . Informix recommends that you make a backup before you convert to Version 7.2x, and then make a second backup once the conversion is complete.
sqlhosts file	The sqlhosts file is mandatory. You must create an sqlhosts file or modify your current sqlhosts file to the format for OnLine Dynamic Server 7.2x.
Utility names	Version 6.0 introduced a new naming convention for the utilities. The INFORMIX-OnLine Dynamic Server utilities have the prefix on (for example, oninit) and the INFORMIX-SE utilities have the prefix se (for example, selog).
sysmaster database	When you initialize OnLine Dynamic Server 7.2x, a script automatically creates the sysmaster database. You must ensure that at least 1100 free pages exist in the root dbspace to build this database.

(1 of 2)

Area of Change	Comments
System resources	The Version 6.0 changes include new requirements for system resources such as shared memory, semaphores, and disk space. When you migrate from OnLine 4.1 or 5.0 to a later version, you must reconfigure the operating-system kernel.
Index requirements	To accommodate new features such as key-value locking, the indexing scheme requires an additional 1 byte of disk space per index-key entry. You must rebuild all user indexes after you migrate from OnLine 4.1 or 5.0. An index on a table with one million records requires approximately 1 additional megabyte of disk space.

(2 of 2)

Changes to Database Utilities

Utility names that began with **tb** in pre-6.0 versions begin with **on** in OnLine Dynamic Server 6.0 and later. For instance, **tbcheck** was replaced by **oncheck**. For the complete list of utilities, refer to the *INFORMIX-OnLine Dynamic Server Administrator's Guide*. Plan to update all references to **tb*** utilities.

You cannot use utilities that use binary formats to load and unload data between OnLine Dynamic Server 7.2x and pre-6.0 versions. To transfer data between OnLine Dynamic Server 7.2x and pre-6.0 versions, you can use only those utilities that load and unload data in ASCII format. For instance, the **tbload**, **tbunload**, and **tbtape** utilities in Version 4.1 and 5.0 use binary data and cannot be used to transfer data to an OnLine Dynamic Server 7.2x database. The **onload**, **onunload**, and **ontape** utilities in OnLine 7.2x use binary data and cannot be used to transfer data to earlier versions. You can use **dbexport** (pre-6.0 version) and **dbimport** (7.2x version) to transfer ASCII data to a Version 7.2x database. You can also use the SQL LOAD and UNLOAD statements to transfer data between versions.

Planning for Migration

This section describes the planning and preparations required by the changes that occurred between Version 5.0 and Version 6.0. The same changes apply to Version 7.2x.

Operating-System Configuration Issues

OnLine Dynamic Server 7.2x requires additional system resources over those required in OnLine 4.1 or 5.0. These resources include additional shared-memory segments, additional semaphores, and additional open-file descriptors per process.

The specific tunable parameters and methods that you use to configure these resources into the operating system vary from platform to platform. For more information, consult the machine-notes file that is installed with your distribution of OnLine Dynamic Server 7.2x and the kernel-configuration instructions for your operating system.

Estimating the Size and Number of Shared-Memory Segments

When you move from Version 4.1 or 5.0, you must recalculate your memory requirements. OnLine Dynamic Server makes more extensive use of shared memory than OnLine 4.1 or 5.0. In addition to housing the buffer cache, OnLine Dynamic Server virtual processors use shared memory to manage user threads and other activities that individual server processes handled in earlier versions. From the standpoint of the operating system, virtual memory that was previously allocated to individual server processes in Version 4.1 and 5.0 is now included in the virtual segment attached by OnLine Dynamic Server.

This new arrangement requires a new method for calculating shared-memory requirements in OnLine Dynamic Server 7.2x that takes the following items into account:

- The *virtual segment*, which is used to manage multiple user threads, data distributions, and other data
- The familiar RSAM or *resident segment*, which is used to manage the buffer cache
- A new *message segment*, which is used to support the shared-memory communication interface



Tip: *Data distributions in OnLine Dynamic Server provide the query optimizer with statistical information about the contents of columns and tables. For information about data distributions, refer to UPDATE STATISTICS in the “Informix Guide to SQL: Syntax.”*

The overall system requirement for user virtual memory includes all three shared-memory segments as well as the space that is needed to hold process images. However, the system requirement for actual physical memory includes only the resident segment and the working sets from other segments and processes. Thus, the physical memory that OnLine Dynamic Server 7.2x requires is proportional to the resident segment, while the requirement for swap space is proportional to the total amount of shared memory that OnLine Dynamic Server uses.

You can use the following steps to generate a rough estimate for the size and number of shared-memory segments that are required for your instance of OnLine Dynamic Server 7.2x:

1. Estimate the total amount of shared memory that is needed to initialize OnLine Dynamic Server 7.2x. You must make separate estimates for each of the three shared-memory segments, as described in the following paragraphs, and add up the total:
 - For an initial estimate of the resident segment size, use the size of shared memory as displayed in the output of **tbmonitor** under OnLine 4.1 or 5.0. Because the resident segment in OnLine Dynamic Server does not include big buffers, you can deduct 4 kilobytes for each 100 buffers in the **BUFFERS** parameter of your OnLine 4.1 or 5.0 **tbconfig** file.
 - Depending on your application, an initial estimate for the virtual segment might be as low as 100 kilobytes per user, or as high as 500 kilobytes per user, plus up to 4 megabytes in addition if you intend to use data distributions. You can obtain an estimate of the number of users under OnLine Dynamic Server 7.2x by adding 12 to the value of the **USERS** parameter in your OnLine 4.1 or 5.0 **tbconfig** file. The initial size of the virtual segment corresponds to the **SHMVIRTSIZE** configuration parameter in the OnLine Dynamic Server 7.2x configuration file.
 - Use the following formula to estimate the size of the message segment:

$$\text{msgseg} = (10,531 * \text{connections}) + 50,000$$

connections is the number of user sessions that can connect through the shared-memory interface. You can set the number of sessions with the **NETTYPE** parameter in the OnLine Dynamic Server 7.2x **ONCONFIG** file.

After you start OnLine Dynamic Server 7.2x, you can obtain a more precise value for **SHMVIRTSIZE** with **onstat -g mem**. You can then reconfigure shared memory more precisely with the actual value for **SHMVIRTSIZE** reported by this command.

2. OnLine Dynamic Server 7.2x can attach additional shared-memory resources during operation when it performs a large sort or other operation that might require more memory than it has previously acquired. To allow OnLine Dynamic Server to expand its use of shared memory while it is operating, reserve a suitable margin of shared memory over that which you estimate is necessary to initialize OnLine Dynamic Server 7.2x. The SHMADD parameter in the ONCONFIG file specifies the size of a dynamically added segment. If you do not specify a value for this parameter in your ONCONFIG file, OnLine Dynamic Server attempts to attach additional shared memory in 8-megabyte segments.

The SHMTOTAL parameter in the ONCONFIG file places an absolute maximum on the amount of shared memory that an instance of OnLine Dynamic Server 7.2x can request. To avoid the risk of exceeding the shared memory provided for a given instance of OnLine Dynamic Server 7.2x, you can set this parameter to indicate the maximum amount of shared memory for that instance. If you set SHMTOTAL to 0 or leave it unassigned, OnLine Dynamic Server continues to attach additional shared memory as needed until no more virtual memory is available on the system.

3. UNIX-specific: Estimate the size and number of shared-memory segments that the operating system needs to provide, and then modify your kernel.

If your operating system *does not* have a segment-size limit, take the following actions:

- a. Set the maximum-segment-size parameter, typically SHMMAX or SHMSIZE, to the total size that is required for OnLine Dynamic Server 7.2x. Include both the amount of memory that is required to initialize OnLine Dynamic Server as calculated in step 1 on page 5-8, and the amount of memory that you wish to allow for dynamic growth as described in step 2 .
- b. Set the operating-system configuration parameter for the maximum number of segments, typically SHMMNI, to at least 1 per instance of OnLine Dynamic Server.

If your system *does* have a segment-size limit, take these actions:

- a. Set the maximum-segment-size parameter for the operating system, typically SHMMAX or SHMSIZE, to the largest value that your system allows.
- b. Use the following formula to calculate the number of segments for your instance of OnLine Dynamic Server 7.2x:

$$\text{SHMMNI} = ((\text{initial_segment}) / \text{SHMMAX}) + \text{dynamic_segments}$$

initial_segment is the segment size that is required to initialize OnLine Dynamic Server 7.2x.

dynamic_segments is the number of segments that you allow to be added during operation of OnLine Dynamic Server 7.2x.

If there is a remainder, round up to the nearest integer value.

4. UNIX-specific: If your operating system uses the SHMSEG configuration parameter to indicate the maximum number of shared-memory segments to which a process can attach, set this system configuration parameter to a value that is equal to or greater than the largest number of segments that you allocated for any one instance of OnLine Dynamic Server 7.2x.

UNIX

Configuring Semaphore Parameters

The operating-system configuration parameters for semaphores are calculated differently for OnLine Dynamic Server 7.1 or later than for pre-6.0 versions of OnLine Dynamic Server. On UNIX systems, the SEMMNI parameter gives the number of semaphore sets. Each instance of OnLine Dynamic Server 7.2x requires one set of semaphores for each group of (up to) 100 virtual processors (VPs) that are initialized with OnLine Dynamic Server, one set for each additional VP that you might add dynamically (while OnLine Dynamic Server is running), and one set for each group of 100 (or fewer) user sessions that are connected through the shared-memory communication interface. Because OnLine Dynamic Server 7.2x utilities such as **onmode** use shared-memory connections, you must configure a minimum of two semaphore sets for each instance of OnLine Dynamic Server 7.2x: one for the initial set of VPs and one for the shared-memory connections used by OnLine Dynamic Server utilities.

The SEMMSL operating-system configuration parameter typically gives the maximum number of semaphores per set; set this parameter to (no less than) 100.

On systems that require you to configure a maximum for the total number of semaphores across all sets, typically given by the SEMMNS operating-system configuration parameter, use the following formula to calculate the total required for each instance of OnLine Dynamic Server 7.2x:

$$\text{SEMMNS} = \text{init_vps} + \text{added_vps} + \text{shmem_users} + \text{concurrent_utils}$$

<i>init_vps</i>	is the number of VPs that are initialized with OnLine Dynamic Server 7.2x. This number includes CPU, PIO, LIO, AIO, SHM, TLI, SOC, and ADM VPs. (For a description of these virtual processors, see the <i>INFORMIX-OnLine Dynamic Server Administrator's Guide</i> .) The minimum value for this term is 15.
<i>added_vps</i>	is the number of VPs that you can add dynamically.
<i>shmem_users</i>	is the number of shared-memory connections that are allowed for this instance of OnLine Dynamic Server 7.2x.
<i>concurrent_utils</i>	is the number of concurrent OnLine Dynamic Server utilities that can connect to this instance. Informix suggests that you allow for a minimum of six utility connections: two for onarchive and four for other utilities such as onmonitor , oncheck , and onstat .

For example, if you start a single instance of OnLine Dynamic Server 7.2x with two CPU VPs and 110 shared-memory users, and you intend to add two CPU VPs dynamically as needed, you must include at least five semaphore sets in the SEMMNI parameter: one set for the initial VPs, two sets for the dynamically added CPU VPs, and two sets for the shared-memory connections. You must set the SEMMSL parameter to at least 100. If your system requires a value for the SEMMNS parameter, you must indicate a total of no less than 133 (15 + 2 + 110 + 6).

If your system uses software packages that require semaphores in addition to the ones that OnLine Dynamic Server 7.2x needs, you must include the total number of semaphore sets that are required by both OnLine Dynamic Server and your other software packages in the SEMMNI parameter. You must set the SEMMSL parameter to the largest number of semaphores per set that any package requires. For systems that require the SEMMNS parameter, you can multiply SEMMNI by the value of SEMMSL to calculate an acceptable value. Or, to arrive at a more precise value for SEMMNS, you can calculate the number of semaphores that are needed for each software package and add those numbers to obtain the total.

Configuring the Number of Open-File Descriptors

Some operating systems require you to specify a limit on the number of file descriptors that a process can have open at any one time. You specify this limit with an operating-system configuration parameter, typically NOFILE, NOFILES, NFILE, or NFILES. The number of open-file descriptors that each instance of OnLine Dynamic Server 7.2x needs is the number of chunks in your database plus the number of network connections that your instance must support. ♦

Operating-System Updates

OnLine Dynamic Server 7.2x might require you to install operating-system updates or *patch releases*. For information about operating-system patches that your installation of OnLine Dynamic Server 7.2x might require, refer to the OnLine Dynamic Server 7.2x machine-notes file.

Disk-Utilization Issues

When you convert from a pre-6.0 version of OnLine Dynamic Server, you must allow for increased disk utilization both during and after the migration process. Some disk resources that must be allocated to the root dbspace during the migration process can be freed for other uses after the process completes. The additional disk space that is required falls into the following categories:

- Space for the conversion processes
- Space in each index entry
- Space for data distributions

The initial requirements during conversion include 1100 additional pages in the root dbspace over its size under pre-6.0 versions of OnLine Dynamic Server. OnLine Dynamic Server uses these added pages to build the **sysmaster** database. You must also provide additional space in the root dbspace for automatic conversion of system catalog indexes.

An additional byte per index entry for each user-table index must be allocated to the dbspace in which each corresponding table resides.

If you intend to use data distributions, you must provide enough temporary space to hold the largest table for which you intend to establish a distribution. For information about data distributions, refer to *UPDATE STATISTICS* in the *Informix Guide to SQL: Syntax*.

The remainder of this section outlines the additional hard-disk requirements for migration to OnLine Dynamic Server 7.2x. You need to examine carefully the amount of disk space left in each dbspace.

The first time that you start up OnLine Dynamic Server 7.2x, the database server performs the following tasks automatically:

- Conversion of system catalog indexes to OnLine Dynamic Server 7.2x (occurs at the start of quiescent mode)
- Creation of **sysmaster** database (occurs at the start of on-line mode)

You must provide enough space for these actions to take effect. After OnLine Dynamic Server begins normal operation, you must rebuild the indexes for user tables. You must allocate adequate disk resources for those indexes as well.

Accommodating System Catalog Indexes

The following formula indicates how many additional pages you must provide to accommodate the growth in system catalog indexes for a given database:

$$\text{growth_in_pages} = (\text{Total} * .10) + (\text{Largest} * 1.10)$$

Total is the total number of leaf pages for all system catalog indexes.

Largest is the number of leaf pages in the largest index.

The additional space that is available during the conversion process must include room for the following items:

- A copy of the largest index plus 10 percent
- 10 percent of the current total of system catalog index pages

OnLine Dynamic Server requires this amount of additional space to accommodate those brief periods in which the old and new versions of an index both reside on disk. The additional 10 percent allows for the case in which the largest index also is the last one to be converted.

You can use the following SQL query within DB-Access or INFORMIX-SQL to determine how many added pages the new system catalog indexes require:

```
UPDATE STATISTICS;  
SELECT ((SUM(leaves) * 0.10) + (MAX(leaves) * 1.10)) sci_added  
FROM sysindexes  
WHERE tabid < 100;
```

Important: You must perform this query from OnLine 4.1 or 5.0.



If sufficient space is not already available in the root dbspace, you must allocate additional chunks or move tables to other dbspaces to make room. You can use the **tbstat -d** command to find the number of free pages in the root dbspace. For details, refer to the *INFORMIX-OnLine Administrator's Guide*, Version 4.1 or 5.0.

Accommodating the sysmaster Database

The **sysmaster** database is created in the root dbspace and cannot be moved or redirected. The **sysmaster** database contains *pseudotables* for monitoring and real tables to store backup information for ON-Archive. Creation of the **sysmaster** database requires up to 1100 free pages in the root dbspace.

Locating Temporary Files and Tables

Version 6.0 and later versions of OnLine Dynamic Server let you use the **DBSPACETEMP** configuration parameter or the **DBSPACETEMP** environment variable to specify the location of temporary files and tables in either raw or cooked space. (Previous versions of OnLine Dynamic Server created temporary tables in the root dbspace by default. Version 6.0 and later versions of OnLine Dynamic Server do not use the **DBPATH** environment variable to locate sort files.) If your computer has at least two hard disks, you might consider mirroring your root dbspace and redirecting the temporary table creation elsewhere. This setup prevents hard-disk failures on the root dbspace from affecting your day-to-day business activities.

Accommodating Data Distributions

If you intend to use data distributions, you must provide adequate space for them in the dbspace that contains the system catalog tables for each database in which they are used. (Use the UPDATE STATISTICS statement, described in the *Informix Guide to SQL: Syntax*, to create data distributions.) The following formula gives the maximum amount of space in bytes that might be required for a data distribution on an individual column. Add the results for each column to obtain the total amount of additional space that is needed for the distributions themselves:

$$\text{dist_space} = (\text{ceil}((4 * ((1/d_res) + 1) * (4 + c_len)) / 765) * 1,116) + 2$$

ceil represents a mathematical function that rounds its argument to the next larger integer. On many systems, this function is called *ceil* or *ceiling*.

d_res is the decimal representation of the resolution that is specified in the UPDATE STATISTICS statement. The default resolution for a HIGH-mode data distribution is 0.5 percent, or 0.005 in this formula.

c_len is the length in bytes for the column. A FLOAT column typically contains 8 bytes; a CHAR 20 column contains 20 bytes.



Tip: This formula yields the maximum possible size for a distribution that contains a number of overflow entries. A typical distribution with few or no overflow entries uses only 25 percent of the maximum space that this formula projects.

For MEDIUM-mode distributions, you must also provide sort space in the dbspace equivalent to 3,000 rows of the widest table. For HIGH-mode distributions, you must provide space for a complete copy of the largest table for which you want a HIGH-mode data distribution.

For example, if you intend to use a HIGH-mode data distribution with the default resolution of 0.5 percent on a CHAR 20 column, the following calculation shows the maximum space that is needed for that distribution in bytes:

$$(\text{ceil}((4 * ((1/0.005) + 1) * (4 + 20)) / 765) * 1,116) + 26$$

This formula works out to 29,042 bytes, as shown by the following calculations:

```
(ceil((4 * (200 + 1) * 24) / 765) * 1,116) + 26
(ceil((4 * 201 * 24) / 765) * 1,116) + 26
(ceil(19,296 / 765) * 1,116) + 26
(ceil(25.223) * 1,116) + 26
(26 * 1,116) + 26
29,016 + 26
```

If this column appears in a table that contains 100,000 rows of 28 bytes each, you must provide a minimum of approximately 2.8 megabytes of sort space in the dbspace to build this distribution.

For a MEDIUM distribution, calculate the required amount of sort space as follows:

```
sort_space = 28B * 3,000 = 84,000B
```

Accommodating User-Table Indexes

The following formula indicates the number of additional pages that are required to accommodate the growth in user-table indexes for a given database:

```
index_growth_pages = (total * 0.10)
```

total is the total number of leaf pages for all user-table indexes.

You might need to add chunks to your existing dbspaces, or perhaps add new dbspaces and move tables to that dbspace to provide additional room. You can use the **tbstat -d** command to find out the number of free pages within the current dbspace. For details, refer to the *INFORMIX-OnLine Administrator's Guide* (Version 4.1 or 5.0).

You can use the following SQL query within DB-Access or INFORMIX-SQL to determine the number of additional pages required by user-table indexes for an entire database:

```
SELECT (SUM(leaves) * 0.1) uti_added
FROM sysindexes
WHERE tabid >= 100;
```

Indexes reside in the same dbspace as the tables to which they refer. Tables can be located in different dbspaces than the databases in which they are managed. If all of your tables reside in the current dbspace, the result of this query indicates the number of pages to add to the database. However, if one or more tables reside in separate dbspaces, you must make sure that those dbspaces include enough room for the new indexes.

To find out the dbspace in which each external table resides, use the following SQL query:

```
SELECT tabname, tabid, (partnum / "0x100000") dbspace_num
FROM systables
WHERE tabid >= 100
AND partnum > 0;
```

For each table that resides in a separate dbspace, use the following SQL query to find out how many additional index pages must be added to the estimate for its dbspace. Add the resulting number of pages to the estimate for that dbspace, and deduct it from your estimate for the current dbspace:

```
SELECT (leaves * 0.1) tbl_added
FROM sysindexes
WHERE tabid = alt_tabid;
```

alt_tabid is the tabid (table ID number) of the table in a separate dbspace as returned by the previous query.

Accommodating the Conversion of User-Table Indexes

The process of converting user-table indexes for use by OnLine Dynamic Server 7.2x is not automatic and requires preliminary planning. You convert user indexes in the last steps of the migration procedure, after OnLine Dynamic Server 7.2x has been installed. For more information, refer to [“Use the oncheck Utility to Convert Indexes” on page 5-35](#).

You can choose from three different methods to convert indexes for a user table. The method that you choose for a given table depends on the size of the table, the degree to which availability of the table is seen as critical, the logging mode of the database, and the time that you can allow for OnLine Dynamic Server 7.2x to stay closed to other users.



You can use the following conversion methods:

- The **oncheck -cI -y** command
- The UPDATE STATISTICS statement
- The DROP INDEX and CREATE INDEX statements

***Tip:** The time needed to convert user-table indexes is proportional to the time required for rebuilding indexes in pre-6.0 versions of OnLine Dynamic Server. Factors that affect the conversion time include the number of indexes per table, the size of each index, the number of available CPUs, and the absence or presence of other user activity on the system.*

The **oncheck -cI -y** command converts indexes while the database is in quiescent mode. This method is preferred for converting large or critical tables. You can also use this method to convert an entire ANSI-compliant database in a single operation. Usually, this method is also the easiest method to use for any table or database. However, OnLine Dynamic Server remains unavailable to other users as long as the **oncheck** utility is running. For more information about the **oncheck** utility, refer to the *INFORMIX-OnLine Dynamic Server Administrator's Guide*.

The UPDATE STATISTICS statement provides added flexibility. You can use it to convert indexes while other users are on-line. However, if you execute this statement within a transaction while other users are working, you risk bringing down OnLine Dynamic Server with a long-transaction error. Handle small tables in ANSI-compliant databases one at a time if other users are working. Do not use this method to convert large or multiple tables in ANSI-compliant databases. For more information about the UPDATE STATISTICS statement, refer to the *Informix Guide to SQL: Syntax*.

Dropping and rebuilding indexes is another conversion option that you can use in place of UPDATE STATISTICS. Because the UPDATE STATISTICS statement also allows you to generate data distributions, it is generally preferred over dropping and rebuilding indexes with the DROP INDEX and CREATE INDEX statements. For more information about these statements, refer to the *Informix Guide to SQL: Syntax*.

To prepare for converting a user index

1. Execute the following query in each database:

```
SELECT tablename
FROM systables
WHERE tabid >= 100
```

2. Determine which method to use for each table in your list.

The following chart gives recommendations for tables depending on their size, importance, the type of database in which they reside, and the urgency with which OnLine Dynamic Server must be brought back on-line. Use **oncheck** to convert large, critical tables. The choice for small-but-critical or large-but-noncritical tables depends on which scenario produces the smaller effect: having OnLine Dynamic Server unavailable, or rebuilding indexes while your system is active.

Type of Table	Needed Quickly	Less Urgent
Large, critical	oncheck -cI -y	oncheck -cI -y
Small, critical	oncheck -cI -y	UPDATE STATISTICS
Large, noncritical	oncheck -cI -y	UPDATE STATISTICS
Small, noncritical	UPDATE STATISTICS	UPDATE STATISTICS



***Tip:** You might be able to use a single method to convert the indexes on all or most tables within a database. Both the **oncheck** command and **UPDATE STATISTICS** statement allow you to use a single command to convert the indexes for an entire database. For details, see the *Informix Guide to SQL: Syntax*.*

Managing Secure-Auditing Log Files

If you intend to use the OnLine Dynamic Server 7.2x secure-auditing features, be advised that OnLine Dynamic Server audit-record log files can grow rapidly to take up a significant amount of space in the file system. Be sure to allow adequate space for the file system in which these files reside. You can configure audit records to minimize the effect of secure auditing on disk utilization. For details, refer to the *INFORMIX-OnLine Dynamic Server Trusted Facility Manual*.

Database System-Configuration Issues

This section provides an overview of database configuration issues involved in the migration process. The following discussions describe only those configuration issues that affect the migration process. For detailed information about OnLine Dynamic Server configuration parameters, refer to the *INFORMIX-OnLine Dynamic Server Administrator's Guide*.

Changing Environment Variables

Version 6.0 and later versions of OnLine Dynamic Server include new environment variables that replace those in OnLine 4.1 and 5.0. Environment variable names that began with **TB** in earlier versions begin with **ON** in OnLine Dynamic Server 6.0 or later. For instance, the **TBCONFIG** environment variable was replaced by the **ONCONFIG** variable. The **SQLEXEC** environment variable is not used in OnLine Dynamic Server 6.0 and later versions.

When you initialize an instance of OnLine Dynamic Server 7.2x, you must set the **INFORMIXSERVER** environment variable to the dbservername of that instance. Applications must also set the **INFORMIXSERVER** environment variable to gain access to databases managed by OnLine Dynamic Server 7.2x.

Revising Configuration Parameter Values

OnLine Dynamic Server 6.0 and later versions recognize new minimum values for certain configuration parameters that existed prior to Version 6.0.

The **BUFFERS** parameter now indicates the maximum number of buffers for disk I/O. To improve performance, increase the **BUFFERS** and **DBSPACETEMP** values, if necessary.

You must increase the value of the **LOCKS** parameter to at least 2,000. OnLine Dynamic Server puts a message in the message log if more locks are needed.

Increasing Logical-Log Space

Make sure that at least 2,000 total log pages are allocated and free for logical logs because the building of the **sysmaster** database requires 1,000 log pages. Informix recommends a 1,000 log-page safety factor. Run **tbstat -l** for your current log-usage status.

Planning for Additional Shared-Memory Usage

The OnLine Dynamic Server architecture, introduced with Version 6.0, combines all the memory used by **sqlturbo** processes in previous versions of OnLine Dynamic Server into a shared-memory section called the *virtual segment* of shared memory. When you migrate from OnLine 4.1 or 5.0, you must allocate enough shared memory with the SHMVIRTSIZE configuration parameter to accommodate the user threads that separate database server processes serviced in pre-6.0 versions. A reasonable initial estimate for SHMVIRTSIZE is 500 kilobytes for each user thread. Additional space might be required for use with data distributions.

OnLine Dynamic Server requires three segments of shared memory as opposed to the one segment that was required in pre-6.0 versions of OnLine Dynamic Server. For information about configuring shared-memory segments in the operating system, refer to [“Estimating the Size and Number of Shared-Memory Segments”](#) on page 5-6.

You use the SHMADD parameter in the ONCONFIG file to set the size of a dynamically added segment. If you do not specify a value for this parameter, OnLine Dynamic Server attempts to attach additional shared memory in 8-megabyte segments.

The SHMTOTAL parameter in the ONCONFIG file places an absolute maximum on the amount of shared memory that an instance of OnLine Dynamic Server 7.2x can request. To avoid the risk of exceeding the shared memory provided for OnLine Dynamic Server, set this parameter to the maximum amount of shared memory that is required for that instance. If you set SHMTOTAL to 0 or leave it unassigned, OnLine Dynamic Server continues to attach additional shared memory as needed until no more virtual memory is available on the system.

Saving Pre-Existing sysmaster Databases

Pre-6.0 versions of OnLine Dynamic Server included the **makeps.sql** script, which created a **sysmaster** database. If this database exists on your system, you can run **dropps.sql** to drop it. If you do not drop the old **sysmaster** database, the conversion process renames it **sysmaster_pre60**.

Configuring Secondary Database Servers for Data Replication

When you configure a secondary database server for use in data replication, the version of OnLine Dynamic Server 7.2x on the secondary host computer must match that of the primary host computer. Chunk names and offsets must also match between instances of OnLine Dynamic Server 7.2x on the primary and secondary hosts. For more information about data replication, refer to the *INFORMIX-OnLine Dynamic Server Administrator's Guide*.

Managing Backups

The ON-Archive backup-management system, introduced in OnLine Dynamic Server 6.0, and table fragmentation, introduced in OnLine Dynamic Server 7.1, might change the placement of databases and tables on disk.

The ON-Archive menu uses a special termcap file that is located in the **tctermcap** file in **\$INFORMIXDIR/etc**. If your terminal is not listed in this file, you might need to add a new entry to use the menu interface. For more information about backup strategies, table organization, and the **tctermcap** file, refer to the *INFORMIX-OnLine Dynamic Server Administrator's Guide*.

Making Performance Comparisons

Informix recommends that you run and record time- and resource-utilization statistics for sample queries and other operations to help you compare performance before and after migration to OnLine Dynamic Server 7.2x. You can compare these statistics with equivalent operations performed after conversion to characterize performance enhancement or degradation. The comparison might help you identify database-configuration parameters that you can adjust to obtain better performance.

Migration from OnLine 4.1 or 5.0 on UNIX to OnLine Dynamic Server 6.0 or Later

The procedure for migrating from a pre-6.0 OnLine to OnLine Dynamic Server 7.2x on UNIX includes the following parts. Each part includes a number of individual steps, as outlined in the sections that follow:

1. Install the latest maintenance release for the current version.
1. Capture configuration and chunk-layout information in OnLine 4.1 or 5.0.
2. Close all transactions and make a final (level-0) backup under OnLine 4.1 or 5.0.
3. Bring OnLine 4.1 or 5.0 off-line.
4. Install and configure OnLine Dynamic Server 7.2x.
5. Reconfigure the operating system.
6. Bring OnLine Dynamic Server 7.2x on-line.
7. Convert user-table indexes.
8. Verify the integrity of the database.
9. Make an initial (level-0) backup under OnLine Dynamic Server 7.2x.

Install the Latest Maintenance Release for the Current Version

Informix recommends that you install the latest maintenance release for your current database server version before you migrate to a new version, especially if you are using Version 5.0. In this scenario, you plan to migrate from OnLine 5.03 to OnLine Dynamic Server 7.2x on UNIX. First, install the latest maintenance release for OnLine, which is 5.07 (as of February 1997), then migrate to Version 7.22. Many minor changes to the 5.x versions are also in the 7.x versions.

Capture Configuration and Chunk-Layout Information

Before you can safely configure OnLine Dynamic Server 7.2x, you must capture the configuration and chunk-layout information under OnLine 4.1 or 5.0. You use this information when you configure OnLine Dynamic Server 7.2x.

To copy database server configuration files

1. Log in as user **informix**.
2. Ask all users of OnLine 4.1 or 5.0 to exit their applications.
3. Rename or make a copy of the configuration file(s) if the current name appears in the following list:
 - **\$INFORMIXDIR/etc/onconfig**
 - **\$INFORMIXDIR/etc/onconfig.std**
 - **\$INFORMIXDIR/etc/sqlhosts**
 - **\$INFORMIXDIR/etc/tbconfig**
 - **\$INFORMIXDIR/etc/tbconfig.std**
 - **\$INFORMIXDIR/etc/tctermcap**
 - **\$INFORMIXDIR/etc/termcap**

You must save these files because the installation procedure for OnLine Dynamic Server 7.2x overwrites them during the installation. Keep the copies available for later use.

To copy chunk-layout and space-use information

Execute the following command to save a listing of your chunk layout and space use in case you need to refer to it:

```
tbstat -d > $INFORMIXDIR/chunk.layout
```

To add chunks or to move tables to accommodate OnLine Dynamic Server 7.2x disk-use requirements

If you need to add chunks or move tables out of the root dbspace to make room for the **sysmaster** database and system catalog indexes, you can do so at this time. You can also add chunks or move tables in this and other dbspaces to accommodate user-table indexes.

For more information about disk-space requirements in OnLine Dynamic Server 7.2x, refer to [“Disk-Utilization Issues” on page 5-13](#). For information about adding chunks or moving tables, refer to the 4.1 or 5.0 version of the *INFORMIX-OnLine Administrator's Guide*.

Close All Transactions and Make a Final Level-0 Backup

Communicate to client users how long you expect the database server to be off-line for the migration. This procedure prepares the transaction log for migration to OnLine Dynamic Server 7.2x and creates a final backup of the database under OnLine 4.1 or 5.0.

To perform an immediate shutdown

Execute the following command to perform an immediate shutdown:

```
tbmode -k
```

Answer **yes** to all the prompts. This step terminates all database server processes that might still be running.

To shut down the system gracefully

1. Execute the **tbmode -sy** command.
2. Wait for all users to exit.
3. Execute the **tbmode -k** command; then answer **yes** to all the prompts.

Make sure the system has shut down completely before you proceed to the next step.

To initiate a fast recovery

Execute the following command to enter quiescent mode and initiate a fast recovery:

```
tbinit -s
```

The **tbinit -s** option rolls forward all committed transactions, rolls back all incomplete transactions since the last checkpoint, and then leaves a new checkpoint record in the log with no open transactions pending. (Refer to the 4.1 or 5.0 version of the *INFORMIX-OnLine Administrator's Guide*.)

To verify the operating mode

Execute the following command to verify that OnLine 4.1 or 5.0 is in quiescent mode.

```
tbstat -
```

The first line of the **tbstat** output contains the status of OnLine 4.1 or 5.0. Figure 5-2 shows that OnLine 4.1 or 5.0 is in quiescent mode.

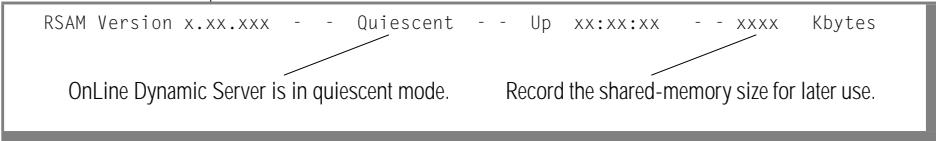


Figure 5-2
*Example of tbstat
Status Line*

To verify the integrity of the data

Before you make a level-0 backup, you might want to verify the integrity of the data.

Use the **tbcheck** utility to verify the integrity of data before you start the backup. You can verify the integrity of the reserve pages, system catalog tables, data, and indexes. (Refer to the 4.1 or 5.0 version of the *INFORMIX-OnLine Administrator's Guide*.) Execute the following commands.

To check	Use the command
Reserve pages	tbcheck -cr
System catalog tables	tbcheck -cc <i>database_name</i>
Data	tbcheck -cD <i>database_name</i>
Indexes	tbcheck -cI <i>database_name</i>



To make a final backup with OnLine 4.1 or 5.0

Use **tbtape** or **tbmonitor** to make a level-0 backup of OnLine 4.1 or 5.0 data. The tape parameters must specify a valid tape device. Be sure to retain and properly label the tape volume that contains the backup. (For more information on making a level-0 tape backup, refer to the 4.1 or 5.0 version of the *INFORMIX-OnLine Administrator's Guide*.)

Important: Backups that you made under OnLine 4.1 and 5.0 are not compatible with the OnLine Dynamic Server 7.2x tape format. Use only the older version of OnLine Dynamic Server to restore the backup tape that you made in this step. You cannot use this backup tape with OnLine Dynamic Server 7.2x.

Bring OnLine 4.1 or 5.0 Off-Line

This procedure shuts down the instance of OnLine 4.1 or 5.0.

To shut down the instance of OnLine

Execute the following command:

```
tbmode -k
```

Answer **yes** to all prompts.

To verify that OnLine 4.1 or 5.0 is off-line

Execute the following command to verify that OnLine 4.1 or 5.0 is in off-line mode:

```
tbmonitor
```



Tip: Use **tbmonitor** instead of **tbstat** to verify the operating mode. The **tbstat** utility is not designed to return the operating-mode status when OnLine Dynamic Server is off-line.

The third line of the DB-Monitor main menu contains the status of OnLine 4.1 or 5.0. The screen shown in Figure 5-3 indicates that OnLine Dynamic Server is off-line. (Refer to the 4.1 or 5.0 version of the *INFORMIX-OnLine Administrator's Guide*.)

```
INFORMIX-OnLine:  Status  Parameters  Dbspaces  Mode  Force-Ckpt
Status menu to view INFORMIX-OnLine
-----Off-Line-----Press CTRL-W for Help -----
```

Figure 5-3
DB-Monitor Main
Menu



OnLine Dynamic Server must be off-line because the older and the newer versions share common files. You cannot install OnLine Dynamic Server if any of the common files are active. Bring OnLine Dynamic Server off-line to ensure that all common files are inactive.

Important: Repeat the previous three sections ([pages 5-25 through 5-28](#)) for each OnLine instance that is to be upgraded.

Install and Configure OnLine Dynamic Server 7.2x

You must be user **root** to install OnLine Dynamic Server 7.2x. Set the **\$INFORMIXDIR** environment variable to the directory where you plan to install OnLine Dynamic Server.

Follow the directions in the *UNIX Products Installation Guide* and the *INFORMIX-OnLine Dynamic Server Administrator's Guide* to install OnLine Dynamic Server 7.2x. The installation script installs OnLine Dynamic Server into the **\$INFORMIXDIR** directory specified for user **root**. The installation script does not bring OnLine Dynamic Server on-line.



Warning: If you install OnLine Dynamic Server 7.2x in the same directory where the older version of OnLine currently resides, the newer version overwrites the older files. If you wish to preserve your product files of earlier versions, you must install OnLine Dynamic Server 7.2x in a different directory.

Before you overwrite the older version, you must take the following precautions:

- If you do not have the original media for the older version, back up the **\$INFORMIXDIR** directory before you install OnLine Dynamic Server 7.2x.
- Copy the configuration file(s) in **\$INFORMIXDIR/etc** to another location on the file system.

When the installation is complete, exit as user **root** and log in as user **informix**.



Important: OnLine Dynamic Server 7.2x includes networking capabilities that are not present in pre-6.0 versions. These capabilities use networking information from configuration files such as the **ONCONFIG** and **sqlhosts** files to establish communications with application processes. You must ensure that these files are present and contain the needed information. To verify these files, establish a connection from DB-Access 7.2x to a test OnLine Dynamic Server instance before you proceed with your installation. If your configuration calls for it, you can also test a remote client-server system.

To set up the configuration file

Copy the standard 7.2x configuration file to the name of your working configuration file, and then edit it. For example, if your configuration file name is **onconfig.1**, you might use the following commands:

```
cd $INFORMIXDIR/etc
cp onconfig.std onconfig.1
vi onconfig.1
```

Refer to the backed-up copy of the OnLine 4.1 or 5.0 configuration file that you saved in [“Capture Configuration and Chunk-Layout Information” on page 5-25](#). Copy the values from matching parameters in the backed-up file to your working configuration file (for example, **onconfig.1**). Modify those values to conform with revised minimums for 7.2x as described in [“Revising Configuration Parameter Values” on page 5-21](#).

To set environment variables for OnLine Dynamic Server 7.2x

At this point, you must set up the environment variables that are needed for communication by both the client applications and OnLine Dynamic Server 7.2x. Make sure that the **INFORMIXDIR** and **PATH** environment variables were set during the installation. (For more information, refer to the *INFORMIX-OnLine Dynamic Server Administrator's Guide*.)

Set the **ONCONFIG** environment variable to the name of the OnLine Dynamic Server 7.2x configuration file, as shown in the following examples:

C shell: `setenv ONCONFIG onconfig.1`

Bourne shell: `NCONFIG=onconfig.1;
export ONCONFIG`

Set the **INFORMIXSERVER** environment variable for all users that need to access OnLine Dynamic Server 7.2x. Set this variable to the name that is listed in the **sqlhosts** file, and in the **DBSERVERNAME** or **DBSERVERALIASES** parameter of the OnLine Dynamic Server 7.2x configuration file, as follows:

C shell: `setenv INFORMIXSERVER dbserver1`

Bourne shell: `INFORMIXSERVER=dbserver1;
export INFORMIXSERVER`

If you use a dbspace or file to store temporary tables, set the **DBSPACETEMP** configuration parameter or **DBSPACETEMP** environment variable to the name of the dbspace or full pathname of that file.

To modify the **sqlhosts** file

OnLine Dynamic Server 7.2x requires an **sqlhosts** file. An **sqlhosts.demo** file has been included in **\$INFORMIXDIR/etc** as an example of the setup that is required for OnLine Dynamic Server 7.2x. If you do not have an **sqlhosts** file already, the installation program renames the **sqlhosts.demo** file to **sqlhosts**. The **sqlhosts** file should include an entry with the following information for each instance of OnLine Dynamic Server 7.2x:

```
dbservername  nettype  hostname  service_name
```

You must modify the entries in this file to support your configuration. For more information on how to modify the **sqlhosts** file, refer to the *INFORMIX-OnLine Dynamic Server Administrator's Guide*.

Important: The client application looks for the **sqlhosts** file in the **\$INFORMIXDIR/etc** directory. However, you can use the **INFORMIXSQLHOSTS** environment variable to change the location or name of the **sqlhosts** file.



Reconfigure the UNIX Operating System

For this step you need to reconfigure your operating system based on the estimates that you determined in “[Planning for Migration](#)” on page 5-6. You need to provide additional shared memory, additional semaphores, and possibly additional hardware resources such as disk drives. To reconfigure the operating system, follow the directions in the machine-notes file included in your OnLine Dynamic Server 7.2x distribution and the kernel-configuration instructions for your operating system.

Bring OnLine Dynamic Server 7.2x On-Line

This step brings OnLine Dynamic Server to quiescent mode. The success of this step depends on adequate operating-system and disk resources, as discussed in [“Planning for Migration” on page 5-6](#). If the system is not brought up in quiescent mode, you get the following error when you attempt to initialize OnLine Dynamic Server 7.2x, and the database server goes off-line:

```
Open transaction detected when changing log versions.
```

To bring OnLine Dynamic Server 7.2x from off-line to quiescent mode

To bring OnLine Dynamic Server 7.2x from off-line to quiescent mode, execute the following command:

```
oninit -s
```

At this point, the system automatically creates the **sysmaster** database. If creation of this database fails, the root dbspace might not include the additional 550 pages that are needed for conversion. Return to [“Install and Configure OnLine Dynamic Server 7.2x” on page 5-29](#), provide enough space in the root dbspace, and repeat the steps in this section. If the **sysmaster** database does not already exist, OnLine Dynamic Server keeps trying to create it each time that you start OnLine Dynamic Server.

Execute the **onstat -m** command to check the message log for the status of the mode change and creation of the **sysmaster** database.



Warning: The logical logs continue to fill with the transactions that result from the creation of the **sysmaster** database. If you run out of log space before the creation of the **sysmaster** database is complete, OnLine Dynamic Server halts with a long-transaction error. Thus, you must back up the logical logs.

Execute the **ontape -a** or **ontape -c** command. Once the logical logs are backed up, processing resumes.

When the system reaches quiescent mode successfully, you can note whether you need to adjust the shared memory and semaphore values for your operating system, which you configured in [“Reconfigure the UNIX Operating System” on page 5-31](#). Check your OnLine Dynamic Server message log for status messages that pertain to the change to quiescent mode. If the message log shows problems, repeat these sections to provide appropriate values for these OnLine Dynamic Server and system resources before you proceed to the next step.

To bring OnLine Dynamic Server 7.2x from quiescent to on-line mode

To change your OnLine Dynamic Server mode from quiescent to on-line, execute the following command:

```
onmode -m
```

At this point, the system attempts to rebuild system catalog indexes. If you try to access them at this time, you might find some of them locked. If these catalogs cannot be rebuilt, you might need to allocate more space in your root dbspace. Along with the 1100 extra pages that are needed for creating the **sysmaster** database, you must add the number of additional pages given by the formula in [“Accommodating System Catalog Indexes” on page 5-14](#). Return to this section, and allocate sufficient space.

Convert User-Table Indexes

If you have not already done so, use the formula provided in [“Accommodating User-Table Indexes” on page 5-17](#) to allocate the additional disk space that is needed for user-table indexes in OnLine Dynamic Server 7.2x. You can now begin converting user-table indexes.

Converting user-table indexes requires planning. You can use several methods to convert indexes for various tables. Depending on the size and demand for a table, you might prefer one method over another. If you have not already done so, generate a list of tables in each of your databases and identify the appropriate conversion method for each index, as described in [“Accommodating the Conversion of User-Table Indexes” on page 5-18](#).

Before you make OnLine Dynamic Server available to regular users, execute the **oncheck -cI -y** command to convert indexes for large or critical tables, as described in [“Use the oncheck Utility to Convert Indexes” on page 5-35](#). You can defer smaller and less-critical tables until after OnLine Dynamic Server is brought on-line. You can then convert these tables through SQL with the UPDATE STATISTICS statement, or the DROP INDEX and CREATE INDEX statements.

You might be able to expedite the process of converting indexes by taking advantage of enhanced features that are provided with OnLine Dynamic Server 7.2x, such as support for multiple CPUs, parallel index builds, and so on. For more information, refer to the *INFORMIX-OnLine Dynamic Server Administrator's Guide*.

The new indexes will be larger than the older indexes no matter which method you choose. Make sure that you allocate enough room in each dbspace for the new indexes. Otherwise, you might need to move tables to other dbspaces before you convert indexes, as described in the following section, “Move Tables to Another Dbspace.”

Move Tables to Another Dbspace

If you neglected to make the proper adjustments to your dbspaces as described under [“Accommodating User-Table Indexes” on page 5-17](#), you might need to move one or more tables to another dbspace to accommodate the larger index in OnLine Dynamic Server 7.2x. If you know that you have enough room for your new indexes, you can skip ahead to [“Use the oncheck Utility to Convert Indexes” on page 5-35](#).

You can use either of the following techniques to move tables during the conversion process:

- The INSERT statement
- The **onunload** and **onload** utilities

To use INSERT to move tables to a new dbspace

1. Create a table in the new dbspace with a temporary name.
2. Use INSERT INTO *newtab* SELECT * FROM *oldtab*.
3. Drop the old table.

4. Create all the indexes in the new table.
5. Rename the new table to the original name.

To use **onunload/onload** to move tables to a new dbspace

1. Create a new table in another dbspace.
2. Drop all the indexes from the old table.
3. Use **onunload** to copy the old table to a file.
4. Drop the table, then update the database to create a new version of the table in the new dbspace.
5. Use **onload** to load the data back into the table.
6. Create all the indexes in the new table.

*Use the **oncheck** Utility to Convert Indexes*

The **oncheck** utility can be helpful when you convert ANSI-compliant databases, and in situations where critical tables must be made accessible as soon as possible.

You can run **oncheck -cI -y** to convert indexes for a single table or an entire database when OnLine Dynamic Server 7.2x is in quiescent mode. After you make sure that no users are present, use **onmode -s** to bring the system into quiescent mode.

To convert indexes for a single table

```
oncheck -cI -y database_name:table_name
```

database_name is the name of the database.

table_name is the name of the table.

To convert indexes for the entire database

```
oncheck -cI -y database_name
```

The **oncheck** command displays messages of the following form:

```
Index index-name is bad. OK to repair it?
```

This message indicates that the existing (4.1 or 5.0) index is not in the correct format for 7.2x. The `-y` option automatically answers `yes` to this prompt, allowing OnLine Dynamic Server 7.2x to convert each index in turn automatically.

Use UPDATE STATISTICS to Convert Indexes



Tip: The `UPDATE STATISTICS` statement includes new options that were introduced in OnLine Dynamic Server 6.0. For details, refer to the “*Informix Guide to SQL: Syntax.*”

You can execute an `UPDATE STATISTICS` statement to convert the indexes for a single table or, in some circumstances, for an entire database. The OnLine Dynamic Server 7.2x `UPDATE STATISTICS` statement automatically converts 4.1 and 5.0 indexes, provided that the dbspace has enough space. For more information, see “[Move Tables to Another Dbspace](#)” on page 5-34.

This method is especially useful for nonlogging databases. Or you can use `UPDATE STATISTICS` in a database without ANSI logging by issuing the statement *outside of a transaction*. When you take the proper precautions, you can even use this method within a transaction or an ANSI-compliant database.



Warning: If you execute an `UPDATE STATISTICS` statement within a transaction, you must follow up with a `COMMIT WORK` or `ROLLBACK WORK` statement to close the transaction. Otherwise, the OnLine Dynamic Server 7.2x instance eventually halts with a long-transaction error.

When you convert indexes within a transaction or an ANSI-compliant database, Informix recommends that you use one of the following approaches to limit the risk of encountering a long transaction:

- Limit the scope of each `UPDATE STATISTICS` statement to a single table. This approach reduces your risk of encountering a long transaction. Execute separate `UPDATE STATISTICS` statements for each table, followed by separate `COMMIT WORK` statements.
- Update statistics for an entire database only when no other users have access to the current instance of OnLine Dynamic Server 7.2x.
- Use **oncheck** to convert indexes while OnLine Dynamic Server is in quiescent mode. For more information, see “[Use the oncheck Utility to Convert Indexes](#)” on page 5-35.

Use **DROP INDEX** and **CREATE INDEX** to Convert Indexes

If neither of the previous two methods seem suitable, you can drop and rebuild the indexes for your tables individually. When you drop and rebuild indexes, you can override the default fill factor that is specified in the **FILLFACTOR** parameter of the **ONCONFIG** file. For details, refer to the *Informix Guide to SQL: Syntax*.

Verify the Integrity of the Data

Informix recommends that you verify the integrity of your data before you run the reversion utility. Choose **Status → Databases** from **ON-Monitor** to get a list of the databases on your database server. Figure 5-4 lists the **oncheck** commands for verifying data integrity.

Action	oncheck Command
Check reserve pages	<code>oncheck -cr</code>
Check extents	<code>oncheck -ce</code>
Check system catalog tables	<code>oncheck -cc database_name</code>
Check data	<code>oncheck -cD database_name</code>
Check indexes	<code>oncheck -cI database_name</code>

Figure 5-4
Commands for
Verifying Data Integrity



Tip: You might see the following warning. It merely means that you have not defined any synonyms.

```
WARNING: No syssytable records found.
```

Make an Initial Backup Under OnLine Dynamic Server 7.2x

Use your OnLine Dynamic Server 7.2x backup tool (**ON-Bar**, **ON-Archive** or **ontape**) to make a level-0 backup. Do not overwrite the tapes you used earlier when you made your final backup of OnLine. For more information, refer to the *INFORMIX-OnLine Dynamic Server Archive and Backup Guide* or the *INFORMIX-OnLine Dynamic Server Backup and Restore Guide*.

Important: Do not restore the backed up logical logs from OnLine 4.1 or 5.0 to the newer version of OnLine Dynamic Server.



Migration Complete

When you finish the level-0 backup, the migration process is complete and users can use OnLine Dynamic Server to access data safely.

Once you successfully migrate to OnLine Dynamic Server 7.2x, you might want to seek ways to obtain maximum performance. If you created sample queries for comparison, you can use them to characterize the performance differences between your earlier version and OnLine Dynamic Server 7.2x. The results of these comparisons might suggest adjustments to configuration parameters, or to the layout of databases, tables, and chunks. For details on topics related to performance, refer to the *INFORMIX-OnLine Dynamic Server Performance Guide*.

GLS

Global Language Support

When you complete the migration from OnLine 4.1 or 5.0 to OnLine Dynamic Server 7.2x, your databases use the default locale, U.S. English. When OnLine Dynamic Server 7.2x is running successfully, you can take the following steps to migrate a database to a non-default locale.

To convert a database to a non-default locale

1. Use the UNLOAD statement or the **dbexport** utility to unload data from the database to be converted.
2. Set the **CLIENT_LOCALE** and **DB_LOCALE** environment variables to support the new GLS locale.
3. Create a database with the new locale by issuing the CREATE DATABASE statement from an application that has the proper client locale variables set in its environment.
4. Modify the schema representation to replace CHAR and VARCHAR columns with NCHAR and NVARCHAR columns, respectively.
5. Use the LOAD statement or the **dbimport** utility to load the data into the new database.

For more information about the GLS feature, refer to [Chapter 9, “Changing Locales”](#) and to the *Guide to GLS Functionality*. ♦

Reverting to OnLine 5.0

This section describes the process for reverting from OnLine Dynamic Server, Version 7.2x to OnLine, Version 5.0. These steps also apply if you are reverting from Versions 7.1UD1, 7.1, or 6.0. Before you can revert from OnLine Dynamic Server to OnLine 5.0, you must modify the configuration limits and remove constructs that OnLine 5.0 does not support.

The procedure for reverting to OnLine includes the following steps, which are described in the next sections:

1. Save copies of your OnLine Dynamic Server configuration files.
2. Remove OnLine Dynamic Server users.
3. Verify the integrity of the data.
4. Back up OnLine Dynamic Server data.
5. Remove features added by later versions.
6. Execute the reversion utility (**onmode -b**).
7. Prepare the TBCONFIG configuration file.
8. Reset environment variables.
9. Bring up OnLine 5.0.
10. Verify the integrity of the data.
11. Back up OnLine 5.0.
12. Bring OnLine 5.0 into on-line mode.

Save Configuration Files

Before you start the reversion process, save copies of your configuration files.

Remove All Users from OnLine Dynamic Server

Remove all users from OnLine Dynamic Server before you begin the reversion process. Warn the users that you plan to shut down OnLine Dynamic Server, and then execute the following command:

```
onmode -s
```

The **-s** flag on **onmode** restricts new access to OnLine Dynamic Server but allows current processing to finish. When all processing is finished, OnLine Dynamic Server goes to quiescent mode, and you can continue the reversion process.

Verify the Integrity of the Data

Before you allow users to access the databases, use the **oncheck** utility to verify that no data was corrupted in the migration process. You can verify the integrity of the reserve pages, system catalog tables, data, and indexes. For more information, see [“Verify the Integrity of the Data” on page 5-37](#).

Back Up OnLine Dynamic Server Data

Informix recommends that you use **ontape** or ON-Archive to make a level-0 backup. For details on making backups, refer to the *INFORMIX-OnLine Dynamic Server Archive and Backup Guide*.

Remove Features Introduced by Later Versions of OnLine Dynamic Server

Before you can revert to an earlier version of OnLine Dynamic Server, you must remove features that were introduced by later versions of OnLine Dynamic Server:

- Remove NLS and GLS databases.
- Disable data replication.
- Change all fragmented tables back into unfragmented tables. The *INFORMIX-OnLine Dynamic Server Administrator's Guide* includes instructions for defragmenting tables.
- Disable role separation.
- Remove unsupported SQL statements.
- Adjust configuration parameters.

Figure 5-5 and Figure 5-6 show parameter limits for Version 5.0 and Version 4.1.

Figure 5-5
Configuration Limits for Version 5.0

Resource	Parameter	5.0 Limit
Logical logs	LOGFILES, LOGSMAX	<= 1 page of entries
Dbspaces	DBSPACES	<= 40
Chunks	CHUNKS	<= 1 page of entries
Buffers	BUFFERS	<= 512 kilobytes

Figure 5-6
Configuration Limits for Version 4.1

Resource	Parameter	4.1 Limit
Logical logs	LOGFILES, LOGSMAX	<= 1 page of entries
Dbspaces	DBSPACES	<= 40
Chunks	CHUNKS	<= 1 page of entries
Buffers	BUFFERS	<= 32,000

The chunk limit depends on the length of the pathname that you chose for the chunk. The limit could range from 13 to 58.

Tip: Logical-log files of differing sizes that were created after 6.0 initialization do not impair the reversion.

Execute the Reversion Utility

To restore OnLine Dynamic Server data to a form that is compatible with OnLine 5.0, execute the following command:

```
onmode -b 5.0
```

The reversion utility includes an implicit **onmode -yuk**. This command forcibly removes all users. After the reversion is complete, the system is in off-line mode.

Tip: The **onmode -b** command also rebuilds the user-table indexes automatically.



Prepare the TBCONFIG Configuration File

Modify the configuration file to eliminate parameters that OnLine 5.0 does not recognize. You might find it easiest to compare your saved configuration file with the **tbconfig.std** file and make adjustments accordingly.

Be sure to include the USERS configuration parameter, which was replaced by USERTHREADS in Version 6.0 and later.

Important: Use the same values for *ROOTPATH*, *ROOTSIZE*, and *ROOTOFFSET* for both versions of OnLine Dynamic Server. *ROOTOFFSET* must equal 0.

Reset Environment Variables

Reset the environment variables to values that are appropriate for the selected version of OnLine Dynamic Server. Remember that in Version 5.0 and 4.1, you specify the configuration file with the **TBCONFIG** environment variable instead of **ONCONFIG**.

The **PDQPRIORITY** environment variable and the SQL **SETPDQPRIORITY** statement were introduced after Version 6.0. If you set the **PDQPRIORITY** environment variable, it does not cause problems but it might cause confusion. You must remove the **SETPDQPRIORITY** statement from your applications.

Bring Up the Desired Version of OnLine Server

Bring up OnLine 4.1 or 5.0 from off-line to quiescent mode. Execute the following command:

```
tbinit -s
```

Verify the Integrity of the Data

Before you allow users to access the databases, check the integrity of the data. Figure 5-7 lists the **tbcheck** commands for verifying data integrity.

Action	oncheck Command
Check reserve pages	tbcheck -cr
Check system catalog tables	tbcheck -cc database_name
Check data	tbcheck -cD database_name
Check indexes	tbcheck -cI database_name

Figure 5-7
*Commands for
Verifying the Data
Integrity*

When you run **tbcheck**, you might see the following message:

```
OLD pn_bytes != NEW pn_nbytes
```

This message does not require any action on your part. It merely indicates that a later version of OnLine Dynamic Server accessed your database.

Back Up OnLine 5.0 Server Data

After you complete the reversion, Informix recommends that you make a level-0 backup. Use the **tbtape** utility to prepare backups. For details on making backups, refer to the appropriate *INFORMIX-OnLine Administrator's Guide*.

Return OnLine 5.0 to On-Line Mode

To change your OnLine 5.0 mode from quiescent to on-line, execute the following command:

```
tbmode -m
```

The reversion is now complete, and users can access the converted data.

Migrating OnLine Workgroup Server

How to Use This Chapter	6-3
OnLine Workgroup Server Differences	6-3
Using Different Server and Client Configurations	6-4
OnLine Workgroup Server 7.22 and Command Center 7.22	6-4
OnLine Workgroup Server 7.12 and Command Center 7.22	6-4
OnLine Workgroup Server 7.22 and Command Center 7.12	6-4
OnLine Workgroup Server 7.12 and Command Center 7.12	6-4
Preparing to Migrate Between Versions	6-5
Check Available Space and System Requirements	6-6
Save a Copy of the Current Configuration Files.	6-6
Place the Database Server in Administration Mode	6-7
Verify the Integrity of the Data	6-8
Back Up Your OnLine Workgroup Server Data	6-9
Preparation Complete	6-9
Upgrading OnLine Workgroup Server to Version 7.2x.	6-9
Bring OnLine Workgroup Server Off-Line	6-10
Install the Database Server and Administration Tools (Windows NT).	6-10
Install the Database Server (UNIX)	6-11
Customize the Database Server Environment	6-13
Configure the Database Server to Use Workgroup Replication	6-13
Bring OnLine Workgroup Server On-Line	6-14
Verify the Integrity of the Data	6-14
Make a Complete Backup of OnLine Workgroup Server.	6-15
Run Update Statistics.	6-15
Verify the Access Path of Your SQL Statements	6-15
Migration Complete	6-16

Migrating to a GLS Locale	6-16
Changing Database Server Definitions	6-16
Reverting OnLine Workgroup Server and Administration Tools . . .	6-17
Remove Unsupported SQL Features	6-17
Uninstall Workgroup Replication Manager (Version 7.22 Only) . .	6-18
Run the Reversion Utility	6-18
Uninstall OnLine Workgroup Server and Administration Tools (Windows NT)	6-19
Remove GLS Features	6-19
Modify Configuration Parameters	6-20
Configuration Changes for Version 7.12	6-20
Configuration Changes for Version 7.22	6-20
Reset Environment Variables	6-21
Environment Variable Changes for Version 7.12	6-21
Reinstall the Earlier Version of OnLine Workgroup Server (Windows NT)	6-22
Reinstall the Earlier Version of OnLine Workgroup Server (UNIX)	6-22
Bring OnLine Workgroup Server On-Line.	6-23
Verify the Integrity of the Data	6-23
Back Up the OnLine Workgroup Server Data	6-23
Reversion is Now Complete	6-23

T

his chapter describes the steps for migrating between Version 7.2x of INFORMIX-OnLine Workgroup Server and earlier versions. The term “Version 7.2x” refers to versions 7.2, 7.21, or 7.22.

How to Use This Chapter

This chapter covers the following topics:

- Using different configurations of OnLine Workgroup Server and the administration tools
- Preparing to migrate
- Upgrading OnLine Workgroup Server and the administration tools
- Reverting OnLine Workgroup Server to Version 7.12

***Tip:** For information on moving OnLine Workgroup Server data between UNIX and Windows NT environments, refer to [Chapter 8, “Moving Between Database Servers.”](#)*



OnLine Workgroup Server Differences

OnLine Workgroup Server for UNIX is at Version 7.2. It supports GLS and uses Version 7.12 of the INFORMIX-Command Center. ♦

OnLine Workgroup Server for Windows NT is at Version 7.22. It supports GLS, ON-Bar, Workgroup Replication, the Gateway products, and uses Version 7.22 of the Command Center. ♦

UNIX

NT

Using Different Server and Client Configurations

This section discusses the effect of different database server and client configurations. Informix supports the following configurations:

- OnLine Workgroup Server 7.22 and Command Center 7.22
- OnLine Workgroup Server 7.12 and Command Center 7.22
- OnLine Workgroup Server 7.22 and Command Center 7.12
- OnLine Workgroup Server 7.12 and Command Center 7.12
- OnLine Workgroup Server 7.2 and Command Center 7.12 ♦

OnLine Workgroup Server 7.22 and Command Center 7.22

Scenario: Upgrade the database server and administration tools. This configuration supports all of the new database server and administration tool functions. If you select **Informix Neighborhood** in the Command Center, the details view displays the version number and type of the database server.

OnLine Workgroup Server 7.12 and Command Center 7.22

Scenario: Upgrade the administration tools but not the database server. With the 7.12 database server, you can use all of the Version 7.22 administration tools except Backup and Restore.

OnLine Workgroup Server 7.22 and Command Center 7.12

Scenario: Upgrade the database server but not the administration tools. This scenario could occur when your site has several client computers to upgrade over a period of time. Users would be able to use all of the Version 7.22 administration tools but not GLS on the clients.

OnLine Workgroup Server 7.12 and Command Center 7.12

Scenario: You decide not to upgrade the database server or the administration tools.

Preparing to Migrate Between Versions

These instructions assume that you are using OnLine Workgroup Server 7.12 on either UNIX or Windows NT. When you migrate OnLine Workgroup Server, Informix suggests that you follow these guidelines:

- Review the release notes for all versions of OnLine Workgroup Server for information about new features, installation, and fixes to problems. Modify applications as needed. Check the documentation notes for information about features not covered in the manuals.

To access the release notes, click the **Release Notes** icon in the **Informix Administration Tools** program group. The release notes and documentation notes are in the %INFORMIXDIR%\release directory (Windows NT) or the \$INFORMIXDIR/release/ows/en_us/0333 directory (UNIX).

- Check the machine notes for special actions required to configure and use OnLine Workgroup Server on your operating system. ♦
- Retain both versions of the Informix product software on disk (if you have enough disk space).
- Retain the installation tapes from both versions of the Informix product software.
- Make a level-0 backup of the database server before and after migration.
- Use a test instance of the database server to test the installation and migration procedures. Before you attempt to convert the production database, use a test instance in the desired communication mode to practice bringing the new database server on-line.

Before you upgrade, migrate, or revert OnLine Workgroup Server, complete the following steps, which are described in the next sections:

1. Check available space and system requirements.
2. Save a copy of the current configuration files.
3. Close all transactions and place the database server in administration mode.
4. Verify the integrity of the data.
5. Back up your OnLine Workgroup Server data.

UNIX

Check Available Space and System Requirements

Before you install OnLine Workgroup Server and the administration tools, verify that your system meets the minimum space and hardware requirements. OnLine Workgroup Server runs on Windows NT 3.51 and Windows NT 4.0 on an NTFS drive. INFORMIX-Command Center and Relational Object Manager run on Windows NT 3.51, Windows NT 4.0, and Windows 95 on either a FAT or NTFS drive.

For information on the system requirements, refer to the *INFORMIX-OnLine Workgroup Server Administration Tools and Database Server Installation Guide* and the **read_ows.txt** file in Answers OnLine.

Save a Copy of the Current Configuration Files

Save a copy of the current ONCONFIG files, located in the **etc** subdirectory of your installation directory, and the **sqlhosts** information:

- **%ONCONFIG%** ♦
- **\$ONCONFIG, osahosts, sqlhosts, ttermcap, termcap** ♦

Save the following files, located in various subdirectories of your installation directory (this step is optional):

- **adtcfg**, located in the **aaodir** subdirectory
- **adtmasks.***, located in the **dbssodir** subdirectory

NT

UNIX

Place the Database Server in Administration Mode

Communicate to client users how long you expect the database server to be off-line for the migration. Terminate all database server processes, and place OnLine Workgroup Server 7.12 in administration mode.

To place the system in administration mode using the Command Center

1. Select **Administration** from the **Server Mode** list box on the **General** page of the Command Center to put the database server in quiescent (administration) mode.
2. Warn all users that you plan to shut down the database server and wait for them to exit. If users are logged on, the Administration wizard lets you notify users when their sessions are to be disconnected.

UNIX

To place the system in administration mode using the UNIX command line

1. Execute the **onmode -sy** command.
2. Warn all users that you plan to shut down the database server and wait for them to exit. ♦



Warning: User data can be lost or damaged if you interrupt data transactions. If data transactions are interrupted, shut down and restart the database server in administration mode to initiate a fast recovery. A fast recovery rolls forward all committed transactions and rolls back all incomplete transactions since the last checkpoint and then leaves a new checkpoint record in the log with no open transactions pending.

NT

Verify the Integrity of the Data

Use the **oncheck** utility to verify the integrity of data before you make a level-0 (complete) backup. If you find any problems with the data, fix them before you make the backup. You can verify the integrity of the reserve pages, system catalog tables, data, and indexes. First, you need to obtain a list of the databases on your database server.

To obtain a list of the databases on your database server

1. In the **Informix Administration Tools** program group, double-click the **Database Explorer** icon.
2. Click a server to see the databases it contains.

To use oncheck on Windows NT

Open a **Command Line Utilities** window from the **Informix Administration Tools** program group. ♦

Figure 6-1 lists the commands for verifying data integrity.

Action	oncheck Command
Check reserve pages	oncheck -cr
Check extents	oncheck -ce
Check system catalog tables	oncheck -cc <i>database_name</i>
Check data	oncheck -cD <i>database_name</i>
Check indexes	oncheck -cI <i>database_name</i>

Figure 6-1
*Commands for
Verifying the Data
Integrity*

For information on **oncheck**, refer to the *INFORMIX-OnLine Workgroup Server Administrator's Guide*. To learn how to use Database Explorer, refer to *Managing Relational Objects*.

Back Up Your OnLine Workgroup Server Data

Make a complete backup of your OnLine Workgroup Server 7.12 data.

NT

On Windows NT, double-click the **Backup and Restore** icon in the **INFORMIX-OnLine Workgroup Server** program group. This icon is available only on the Windows NT server computer in Version 7.12. You can also use **ontape** to back up on Windows NT. ♦

UNIX

To launch Backup and Restore, enter **\$INFORMIXDIR/bin/bar** at the UNIX prompt. You can also use **ontape** or **ON-Archive** to back up on UNIX. ♦



Warning: *The 7.12 and 7.22 versions of Backup and Restore are not compatible. The 7.12 version uses **ontape**, but the 7.22 version uses **ON-Bar**. Also, backups that you make under older versions of OnLine Workgroup Server are not compatible with Version 7.22. Do not try to restore these backups to OnLine Workgroup Server, Version 7.22.*

For more information about backing up OnLine Workgroup Server 7.12 or 7.2, refer to *Using the OnLine Workgroup Server*. For information about backing up OnLine Workgroup Server 7.22, refer to *Using the INFORMIX-Command Center*.

Preparation Complete

Now you are ready to upgrade, migrate, or revert OnLine Workgroup Server. For information on upgrading, see “Upgrading OnLine Workgroup Server to Version 7.2x.” For information on reverting, see [“Reverting OnLine Workgroup Server and Administration Tools” on page 6-17.](#)

Upgrading OnLine Workgroup Server to Version 7.2x

When you upgrade from OnLine Workgroup Server, Version 7.12 to Version 7.2x, you can install and test Version 7.2x with the same database server name, configuration files, environment variables, and shared-server machine that you used for the earlier version. After you install OnLine Workgroup Server, Version 7.2x, and verify that it works, you might want to modify the configuration files and environment variables to take advantage of new features such as Global Language Support (GLS).

NT

Complete the following migration steps if you are upgrading OnLine Workgroup Server on Windows NT or UNIX:

1. Bring OnLine Workgroup Server off-line.
2. Install the new version of OnLine Workgroup Server.
3. Customize the database server environment. (This step is optional.)
4. Configure the database server for Workgroup Replication. (This step is optional.) ♦
5. Bring OnLine Workgroup Server on-line.
6. Verify the integrity of the data.
7. Make an initial, complete backup of OnLine Workgroup Server.
8. Run UPDATE STATISTICS.
9. Verify the access path of your SQL statements.

Bring OnLine Workgroup Server Off-Line

Shut down OnLine Workgroup Server.

***Tip:** The installation program automatically shuts down the old OnLine Workgroup Server and starts the new OnLine Workgroup Server.*



NT

Install the Database Server and Administration Tools (Windows NT)

You can upgrade the administration tools only, the database server only, or both. If a previous version of the database server or administration tools is on the computer, the **Upgrade** page appears when you install the new product. The Installation wizard replaces the files but does not reconfigure the database server.

To perform the upgrade

1. When the **Upgrade** page appears, click **Next** for the INFORMIX-OnLine Installation wizard. Click **Next** again to begin the installation procedure.
2. Enter the serial number and serial-number key.

3. The installation program automatically verifies and brings down OnLine Workgroup Server, copies the new files, and preserves the database and dbspace data. Then the installation program starts OnLine Workgroup Server, Version 7.22, with the same configuration and shared-server machine.
4. A dialog box displays the following message:

Your INFORMIX-OnLine Workgroup Server is being
launched.

The Installation wizard updates the SQLHOSTS registry keys and OSAHOSTS information automatically.

To install and configure onsnmp

Workgroup Replication and the **onsnmp** utility require Windows NT SNMP. The installation and upgrade program checks the registry for the SNMP master agent. If the master agent has not been installed, the program displays a warning message but does not configure the registry for Workgroup Replication or **onsnmp**. If you later choose to install the SNMP master agent, you must run the `%INFORMIXDIR%\bin\inssnmp.exe` command line utility to install the SNMP subagents. You need not reinstall the database server. ♦

UNIX

Install the Database Server (UNIX)

To upgrade OnLine Workgroup Server for UNIX to Version 7.2, install the new database server files in the same directory as the old database server files. You do not need to reinstall the administration tools because they were not upgraded for Version 7.2.

To launch the installation program

1. Log in as user **root**.
2. Follow the instructions in the *INFORMIX-OnLine Workgroup Server Installation Guide for UNIX* to launch the installation program.

To perform the upgrade

1. Supply customer information. You must enter the serial number and serial-number key. The remaining information is optional, but it is used to create your product registration:
 - On the **Licensing the Software** screen, enter the case-sensitive serial number and serial-number key exactly as shown on the serial-number key card in your software media box.
 - On the **Product Registration** screen, enter your name, title, and company name.
 - On the **Address Information** screen, enter your address.
 - On the **Contact Information** screen, enter your telephone number, fax number, and Internet address.
2. On the **OnLine Workgroup Server Installation Options** screen, choose option 1, **Install**.
3. On the **Run Setup Again** screen, choose option 1, **Install All Files, But Leave Configuration Alone**, as shown in Figure 6-2. This option copies the new database server files without changing the configuration and SQLHOSTS information.

Run Setup Again?

INFORMIX-OnLine Workgroup Server is already set up on this computer. If you want to set up the INFORMIX-OnLine Workgroup Server again, select an option.

1. Install all files, but leave configuration alone.
2. Install all files and reconfigure the product.

Figure 6-2
Run Setup Again

4. The installation program automatically verifies and brings down OnLine Workgroup Server, copies the new files, and saves the database and dbspace data. Then the installation program starts OnLine Workgroup Server, Version 7.2, with the old configuration.

Warning: Do not select option 2, **Install OnLine Workgroup Server and Reconfigure**. If you select option 2, your configuration and database information will be lost. ♦



Customize the Database Server Environment

If you are an advanced user, you can customize the ONCONFIG configuration file and environment variables for OnLine Workgroup Server. Use a text editor to edit the ONCONFIG file. For more information on configuration parameters, refer to the *INFORMIX-OnLine Workgroup Server Administrator's Guide*.



Important: Use the same values for *ROOTOFFSET*, *ROOTSIZE*, and *ROOTPATH* that you used for the earlier version of OnLine Workgroup Server.

To customize environment variables

Use the **setnet32** tool to customize the environment variables on the client. For more information on environment variables, refer to the *Informix Guide to SQL: Reference*.

NT

Configure the Database Server to Use Workgroup Replication

Perform this step only if you plan to use Workgroup Replication with OnLine Workgroup Server for Windows NT 7.22. Before you can activate Workgroup Replication, you need to update the ONCONFIG file, **services** file, and SQLHOSTS registry for OnLine Workgroup Server. To activate Workgroup Replication for the first time, follow these steps:

1. Bring down OnLine Workgroup Server.
2. Define one or more **dbserver aliases** for Workgroup Replication.
3. Define a **group name** for Workgroup Replication.
4. Bring OnLine Workgroup Server on-line.
5. Use the Replication Manager GUI to define a replicate.
6. Start Workgroup Replication.
7. Verify that the SNMP service is installed on the host and that the SNMP subagents are registered.

For complete information on configuring for Workgroup Replication, refer to the *Guide to Informix Workgroup Replication*. For information on how to use SNMP, refer to the *Informix SNMP Subagent Guide*. ♦

NT

Bring OnLine Workgroup Server On-Line

The installation program brings OnLine Workgroup Server on-line automatically.

If you customized the database server environment, bring down and restart OnLine Workgroup Server with the Command Center. When you restart OnLine Workgroup Server, the changes to the configuration parameters and environment variables take effect.

To start OnLine Workgroup Server manually using the Command Center

1. In the **Informix Administration Tools** program group, double-click the **Command Center** icon.
2. In the Command Center, select the database server in the **All Servers** tree view or the **Servers** list box.
3. Choose **Server→On-line**.

For more information, refer to *Using the INFORMIX-Command Center* or the General help. ♦

UNIX

To start OnLine Workgroup Server manually using the Command Center

1. In the **INFORMIX-OnLine Workgroup Server** program group, double-click the **Command Center** icon. In the Command Center, choose a database server from the **Servers** menu.
2. On the **General** page, select **On-Line** in the **Server Mode** list box.

To start OnLine Workgroup Server for UNIX at the command line, enter `oninit` on the server. ♦

Verify the Integrity of the Data

Use the **oncheck** utility to verify the integrity of data, as described in [“Verify the Integrity of the Data” on page 6-8](#).

Make a Complete Backup of OnLine Workgroup Server

Use Backup and Restore to make a complete, whole-system backup of OnLine Workgroup Server. For more information, see “Backing Up and Restoring Data” in *Using the INFORMIX-Command Center* or the General Help.

Run Update Statistics

After you complete the migration procedure, run the UPDATE STATISTICS statement. The UPDATE STATISTICS statement updates the information that OnLine Workgroup Server uses to plan efficient queries. For more information about UPDATE STATISTICS, refer to the [Informix Guide to SQL: Syntax](#).

Verify the Access Path of Your SQL Statements

Use the SET EXPLAIN statement to verify that the access path of your SQL statements did not change when you migrated to OnLine Workgroup Server. If you have SET EXPLAIN output from the source database server, run SET EXPLAIN for OnLine Workgroup Server. Compare the SET EXPLAIN output from both the source and target database servers. SET EXPLAIN writes the access path that the optimizer chooses for each query to the SET EXPLAIN output file. The optimizer chooses the fastest path of execution for table joins.

NT

For Windows NT, the SET EXPLAIN output filename is %INFORMIXDIR%\sqlexpln\<username>.out. ♦

UNIX

The UNIX SET EXPLAIN output filename is **sqexplain.out**. ♦

If the SET EXPLAIN output file shows that a different access path was used, complete the following steps:

1. Check the **OPTCOMPIND** environment variable or configuration parameter.
2. Check the **DBSPACETEMP** environment variable or configuration parameter to ensure that adequate temporary dbspaces are defined. You might need to define more temporary dbspaces.
3. Analyze the query access paths, and modify the schema to improve the performance if necessary.

Migration Complete

The first time OnLine Workgroup Server is brought on-line, the **sysmaster** database is built. Check the message log to ensure that the **sysmaster** database build has completed before you allow users to access the database server. After you ensure that client users can access data on OnLine Workgroup Server, the migration process is complete.

Migrating to a GLS Locale

If you wish to migrate OnLine Workgroup Server to a non-English GLS locale, set the DB_LOCALE and CLIENT_LOCALE environment variables before you open the Version 7.12 database in Version 7.2x. If your previous database server version used Native Language Support (NLS), replace the NLS environment variables, such as COLLCHAR, with GLS environment variables. For information on working with locales and how to set GLS environment variables, see the *Guide to GLS Functionality* and [Chapter 9, “Changing Locales.”](#)



Important: This version of GLS does not support Middle Eastern languages.

Changing Database Server Definitions

The installation program automatically updates the registry information and the database server definitions on the shared-server (SQLHOSTS) machine. The *database server definitions* consist of the database server name, TCP/IP host names, network types, and the service names. Use the **Setup** program if you want to change the **informix** user password or specify a different computer as the shared-server machine. In Version 7.22, **Setup** allows you to edit the database server definitions. For example, you might specify a different shared-server machine if you have migrated the database server to a new computer or connected a single client to several database servers.

For information on how to use **Setup** to configure database servers, see *Using the INFORMIX-Command Center* or the General help. For information on connectivity, see the *INFORMIX-OnLine Workgroup Server Administrator's Guide*.

NT

Reverting OnLine Workgroup Server and Administration Tools

This section describes the steps for reverting from OnLine Workgroup Server and the administration tools from Version 7.22 to Version 7.12.

Follow the preparatory steps, described in [“Preparing to Migrate Between Versions” on page 6-5](#), then complete the following steps:

1. Remove unsupported SQL features.
2. Uninstall Workgroup Replication Manager, if it is installed. ♦
3. Run the reversion utility (**onmode -b**).
4. Uninstall OnLine Workgroup Server and the administration tools.
5. Remove GLS features (only if GLS was used).
6. Modify configuration parameters.
7. Reset environment variables.
8. Reinstall the old version of OnLine Workgroup Server.
9. Bring OnLine Workgroup Server on-line.
10. Verify the integrity of the data.
11. Make an initial, complete backup of OnLine Workgroup Server.

Remove Unsupported SQL Features

Before you revert, you must remove SQL features that the earlier version of OnLine Workgroup Server does not support. See the “New Features of This Product” section in the appropriate version of the *Informix Guide to SQL: Syntax*.

NT



Uninstall Workgroup Replication Manager (Version 7.22 Only)

Skip this section if Workgroup Replication is not installed on your system.

To revert to an earlier version if Workgroup Replication is active

1. Uninstall Workgroup Replication Manager.
2. Stop Workgroup Replication.
3. Execute the **onmode -b** command, as described in “Run the Reversion Utility,” to revert to the earlier version of OnLine Workgroup Server.

Warning: If you try to revert to a previous version of OnLine Workgroup Server while Workgroup Replication is active, the reversion will fail.

To revert to an earlier version if Workgroup Replication is inactive

In this situation, Workgroup Replication was installed but is not active on this database server.

1. Uninstall Workgroup Replication Manager.
2. Execute the **onmode -b** command, described in “Run the Reversion Utility,” to revert to the earlier version.

For more information, see the *Guide to Informix Workgroup Replication*. ♦

Run the Reversion Utility

OnLine Workgroup Server must be running when you execute the reversion utility. Execute the reversion utility to revert the database, where *version_number* is the earlier OnLine Workgroup Server version (see Figure 6-3):

```
onmode -b version_number
```


After the reversion is complete, OnLine Workgroup Server is off-line. For more information about the **onmode -b** command, refer to [Chapter 10, “Utilities for Data Migration.”](#)

Revert from	Revert to	Command
Version 7.2x	Version 7.12	onmode -b 7.1
Version 7.22	Version 7.2	onmode -b 7.2

Figure 6-3
Reverting to an Earlier OnLine Workgroup Server

NT

Uninstall OnLine Workgroup Server and Administration Tools (Windows NT)

You can uninstall the database server only, the administration tools only, or both.

1. Double-click the **Uninstall** icon in the **Informix Administration Tools** program group.
2. In the Uninstall dialog box, check **Remove OnLine Server**.

Warning: Do not check **Remove all OnLine databases, supporting files and all database information**. If you check this option, your configuration and database information will be lost, making reversion impossible.

3. To uninstall the administration tools, check **Remove Administration Tools**.
4. Click **OK** to uninstall OnLine Workgroup Server. ♦

Remove GLS Features

Skip this step if OnLine Workgroup Server uses the default English locale (**en_us.8859-1**). To revert OnLine Workgroup Server from GLS to Native Language Support (NLS), set the appropriate NLS locales and environment variables. For information on working with locales, see the *Guide to GLS Functionality* and [Chapter 9, “Changing Locales.”](#)



NT

Modify Configuration Parameters

Version 7.2 and 7.22 use configuration parameters that did not appear in earlier versions of OnLine Workgroup Server. In addition, the default values of some parameters changed.

Configuration Changes for Version 7.12

You must add the ONLANGMAP configuration parameter to your ONCONFIG file before you start OnLine Workgroup Server 7.12. ONLANGMAP takes the value **en_US-English**. For information on ONLANGMAP, see the INFORMIX-OnLine Dynamic Server *for Windows NT*, Version 7.12.TC2, documentation notes. ♦

If you are reverting to Version 7.12, remove the HETERO_COMMIT configuration parameter from your ONCONFIG file.

Configuration Changes for Version 7.22

You might need to revise the value of the ALARMPROGRAM configuration parameter in your configuration file.

If you used Workgroup Replication and are reverting to Version 7.12 or 7.2, remove the following parameters from your ONCONFIG file:

- CDR_LOGBUFFERS
- CDR_EVALTHREADS
- CDR_DSLOCKWAIT
- CDR_QUEUEMEM
- CDR_NIFUSEHELP
- CDR_NIFMEMS
- SCDR_NIFQUEUES

For more information on these configuration parameters, see the *Guide to Informix Workgroup Replication*.

Reset Environment Variables

Reset the environment variables to values that are appropriate for your version of OnLine Workgroup Server. Use **setnet32** to modify the environment variables. For information on environment variables, refer to the *Informix Guide to SQL: Reference*.

Environment Variable Changes for Version 7.12

OnLine Workgroup Server, Version 7.12, supports NLS, not GLS. When you revert to Version 7.12, delete the following environment variables:

- **CC8BITLEVEL**
- **CLIENT_LOCALE**
- **DBCENTURY**
- **DBFLTMASK**
- **DBONPLOAD**
- **DB_LOCALE**
- **ESQLMF**
- **GLS8BITFSYS**
- **GL_DATES**
- **GL_DATETIME**
- **ONPLOAD**
- **PLCONFIG**
- **SERVER_LOCALE**
- **THREADLIB**

Add the following environment variables:

- **DBNLS**
- **COLLCHAR**
- **LANG**

NT

Reinstall the Earlier Version of OnLine Workgroup Server (Windows NT)

Reinstall the earlier version of OnLine Workgroup Server and administration tools in the same directory as the Version 7.22 files.

To perform the installation

1. On the **Run Installation Again** page, select the **Copy all files, but leave configuration alone** option.
2. Supply your **Serial Number** and **Serial Number Key**, as shown on the serial-number key card.
3. Select one or both of the components you want to install: **OnLine Database Server** or **Administration Tools**. Click **Next**.
4. The installation program automatically copies the 7.12 database server and administration tools files, and saves the database and dbspace data. It starts the earlier version of OnLine Workgroup Server with the old configuration.

***Warning:** You must add the ONLANGMAP parameter to the ONCONFIG file before you start the instance of OnLine Workgroup Server 7.12 that contains the reverted databases. ONLANGMAP takes the value **en_US-English**. The ONLANGMAP configuration parameter ensures backward compatibility with non-NLS locales on Windows NT. If the database server crashes on install, just restart it.*

***Warning:** Do not select **Copy all files and reconfigure the product**. If you select this option, your configuration and database information will be lost. ♦*



UNIX

Reinstall the Earlier Version of OnLine Workgroup Server (UNIX)

Reinstall the earlier version of OnLine Workgroup Server in the same directory as the Version 7.2 files. You do not need to reinstall the administration tools because they were not upgraded for Version 7.2.

To perform the installation

1. Supply customer information.
2. On the **OnLine Workgroup Server Installation Options** screen, choose option 1, **Install**.

3. On the **Run Setup Again** screen, choose option 1, **Install All Files, But Leave Configuration Alone**. This option copies the new database server files without changing the configuration and SQLHOSTS information.
4. The installation program automatically verifies and brings down OnLine Workgroup Server, copies the new files, and saves the database and dbspace data. It starts the earlier version of OnLine Workgroup Server with the old configuration.



Warning: Do not select option 2, **Install OnLine Workgroup Server and Reconfigure**. If you select option 2, your configuration and database information will be lost. ♦

Bring OnLine Workgroup Server On-Line

The installation program brings OnLine Workgroup Server on-line automatically.

Verify the Integrity of the Data

To verify the integrity of your data, follow the steps as described in [“Verify the Integrity of the Data” on page 6-8](#).

Back Up the OnLine Workgroup Server Data

Use Backup and Restore to make a complete backup of OnLine Workgroup Server. For more information, see [“Back Up Your OnLine Workgroup Server Data” on page 6-9](#) or *Using the OnLine Workgroup Server, Version 7.12*.



Important: Do not overwrite the tapes that you used to back up OnLine Workgroup Server, Version 7.2x.

Reversion is Now Complete

After the **sysmaster** database is built, the reversion process is complete. Ensure that client users can access data on the earlier version of OnLine Workgroup Server.

Migrating SE

Importing and Exporting Data	7-3
Migrating Between Different Versions of SE	7-4
Preparing to Migrate from a Pre-6.0 SE to SE 7.2x	7-4
The sqlhosts File	7-4
Environment Variables	7-4
SE Utilities	7-5
Preparing to Migrate from a Post-6.0 SE to SE 7.2x.	7-5
Reverting to an Earlier Version of SE.	7-6
Converting C-ISAM Files to SE.	7-6
Migrating from C-ISAM to OnLine Dynamic Server or OnLine Workgroup Server	7-7

T

his chapter describes how to migrate data between different versions of INFORMIX-SE database servers. [Chapter 8, “Moving Between Database Servers,”](#) covers moving data between SE and OnLine Dynamic Server.

Importing and Exporting Data

You can use the following utilities to import and export data from an SE database server:

- **dbexport/dbimport**
- **UNLOAD/dbschema/LOAD**
- **UNLOAD/dbschema/dbload**

Let your target destination, performance and ease-of-use requirements determine the method that you use. Refer to [Figure 1-6 on page 1-9](#), for details. [Chapter 2, “Moving Between Computers,”](#) describes the procedures for exporting and importing data. [Chapter 10, “Utilities for Data Migration,”](#) describes the syntax of the **dbexport**, **dbimport**, **UNLOAD**, **dbschema**, **LOAD**, and **dbload** utilities.

Migrating Between Different Versions of SE

Unlike OnLine Dynamic Server data, SE data is stored in ordinary UNIX files. The structure of these files has remained the same between versions of SE, so that migrating from one version to a later version requires little preparation. SE manages the file contents, but the operating system manages the I/O.

Preparing to Migrate from a Pre-6.0 SE to SE 7.2x

Version 6.0 of Informix products introduced changes in the way clients connect to database servers. The names of database server utilities also changed.

The sqlhosts File

Pre-6.0 versions of SE did not require an **sqlhosts** file unless you used SE with INFORMIX-NET. Beginning with Version 6.0, Informix products require an **sqlhosts** file to specify connections between clients and servers. For information about preparing your **sqlhosts** file, refer to the *INFORMIX-SE Administrator's Guide*.

Environment Variables

Beginning with Version 6.0, SE requires the following environment variables:

- **INFORMIXDIR**
- **PATH**
- **INFORMIXSERVER**

Depending on your network configuration, you might also need the following environment variables:

- **SQLEXEC**
- **SQLRMDIR**
- **SQLRM**

For information about how to set these environment variables, refer to the *INFORMIX-SE Administrator's Guide*.

SE Utilities

Beginning with Version 6.0, the names of the SE utilities **bcheck** and **dblog** changed to **secheck** and **selog**, respectively. If you have scripts that use these utilities, you must update the names of the utilities.

To migrate from a pre-6.0 version of SE to SE 7.2x

1. Ask all users to exit from their applications.
2. Verify the validity of your data.

For Version 4.1 or 5.x, execute the following command for each table in the database:

```
bcheck tablename
```

For more information about these utilities, refer to the appropriate version of the *INFORMIX-SE Administrator's Guide*.

3. Install SE 7.2x.

For information about installing SE, refer to the *UNIX Products Installation Guide*.

Preparing to Migrate from a Post-6.0 SE to SE 7.2x

When you migrate from a post-6.0 version of SE to SE 7.2x, you do not need to change the environment variables or the **sqlhosts** file.

To migrate from a post-6.0 version of SE to SE 7.2x

1. Ask all users to exit from their applications.
2. Verify the validity of the data.

For Version 6.0 and later, execute the following command for each table in the database:

```
secheck tablename
```

For more information about these utilities, refer to the appropriate version of the *INFORMIX-SE Administrator's Guide*.

3. Install SE 7.2x.

For information about installing SE, refer to the *UNIX Products Installation Guide*.

Reverting to an Earlier Version of SE

Informix does not provide a reversion utility for SE. To move your database to an earlier version of SE, you must unload and then reload your data. Follow the procedures for **dbexport/dbimport** or **UNLOAD** that are described in [Chapter 8, “Moving Data from OnLine Dynamic Server or OnLine Workgroup Server to SE”](#) on page 8-12.

Converting C-ISAM Files to SE

This section describes how to convert C-ISAM files to SE format. C-ISAM files are organized differently than relational tables. C-ISAM files tend to be much larger and are not normalized. Therefore, you need to convert C-ISAM applications to the SQL format that you can use with SE:

1. In SE, use the SQL **CREATE TABLE** statement to create a table that corresponds to the data fields in the C-ISAM application. (Do not use the name of the C-ISAM file for the table name.)
2. Delete the empty **.dat** file that SE created in the **CREATE TABLE** statement. It has the name `tablename|tabid|.dat`. Also delete the empty **.idx** file.
3. Either move the C-ISAM file to the **.dbs** directory, or update the **systables** system catalog with the name of the C-ISAM data file. If you update both the C-ISAM and SE files with logging turned on, you must use common logging.

Migrating from C-ISAM to OnLine Dynamic Server or OnLine Workgroup Server

After you convert the C-ISAM data files to SE format, use **dbexport** and **dbimport** to migrate the data from SE to OnLine Dynamic Server or OnLine Workgroup Server. Follow the procedures that are described in [“Moving Data from OnLine Dynamic Server or OnLine Workgroup Server to SE” on page 8-12](#). For more information on working with C-ISAM files, see the *C-ISAM Programmer's Manual*.

Tip: For details on converting C-ISAM data files to SE, contact your local systems engineer for assistance.



Moving Between Database Servers

Choosing a Migration Method	8-5
Moving Data from SE to OnLine Dynamic Server or OnLine Workgroup Server	8-6
Using the dbexport and dbimport Utilities	8-6
Using the UNLOAD Statement	8-8
Using UNLOAD, dbschema, and LOAD.	8-8
Using UNLOAD, dbschema, and dbload	8-10
Adapting Your Programs for OnLine Dynamic Server or OnLine Workgroup Server	8-11
Moving Data from OnLine Dynamic Server or OnLine Workgroup Server to SE.	8-12
Eliminating Features That SE Does Not Support	8-12
Using the dbexport and dbimport Utilities	8-12
Using the UNLOAD Statement	8-14
Using UNLOAD, dbschema, and LOAD.	8-14
Using UNLOAD, dbschema, and dbload	8-15
Adapting Your Programs for SE	8-16
Moving Data from OnLine Dynamic Server to OnLine Workgroup Server.	8-17
Moving Data in the Same Environment	8-17
Moving Data Between Different Environments.	8-17
Save a Copy of the Current Configuration Files.	8-18
Verify the Integrity of the Data	8-18
Back Up OnLine Dynamic Server	8-19
Export the OnLine Dynamic Server Databases	8-19
Bring OnLine Dynamic Server Off-Line	8-19
Install and Configure OnLine Workgroup Server	8-20
Verify Port Numbers and Services File	8-20
Customize the Database Server Environment	8-20

Bring OnLine Workgroup Server On-Line	8-21
Import the Databases to OnLine Workgroup Server	8-21
Verify the Integrity of the Data	8-22
Back Up OnLine Workgroup Server	8-22
Run Update Statistics	8-22
Migration Complete	8-22
Adapting Your Programs for OnLine Workgroup Server.	8-23
Using the dbexport and dbimport Utilities	8-23
Using the UNLOAD Statement	8-24
Using UNLOAD, dbschema, and LOAD	8-24
Using UNLOAD, dbschema, and dbload.	8-25
Moving Data from OnLine Workgroup Server to OnLine Dynamic Server	8-26
Moving Data in the Same Environment	8-26
Moving Data Between Different Environments	8-27
Save a Copy of the Current Configuration Files	8-28
Verify the Integrity of the Data	8-28
Back Up OnLine Workgroup Server	8-28
Export the OnLine Workgroup Server Databases	8-29
Shut Down OnLine Workgroup Server (UNIX)	8-29
Shut Down OnLine Workgroup Server (Windows NT)	8-29
Install and Configure OnLine Dynamic Server and the Administration Tools (Windows NT)	8-29
Install OnLine Dynamic Server (UNIX)	8-30
Verify Port Numbers and Services File	8-30
Customize the Database Server Environment	8-30
Bring OnLine Dynamic Server On-Line	8-31
Import the Databases into OnLine Dynamic Server	8-31
Verify the Integrity of the Data	8-31
Back Up OnLine Dynamic Server.	8-32
Run Update Statistics	8-32
Migration Complete	8-32
Adapting Your Programs for OnLine Dynamic Server	8-32
Using the dbexport and dbimport Utilities	8-33
Using the UNLOAD Statement	8-34
Using UNLOAD, dbschema, and LOAD	8-34
Using UNLOAD, dbschema, and dbload.	8-35
Moving OnLine Dynamic Server Data Between Environments	8-36
Procedure for Moving Data Between Environments	8-36
Using the Migration Utilities	8-37

Adapting Your Programs for the UNIX or NT Environment	8-37
Migration Complete.	8-37
Moving OnLine Workgroup Server Data Between Environments	8-38
Procedure for Moving Data Between Environments	8-38
Using the Migration Utilities.	8-39
Adapting Your Programs for the UNIX or NT Environment	8-39
Migration Complete.	8-39

This chapter describes the steps for migrating between Informix database servers. [Figure 1-5 on page 1-8](#) shows the paths for migrating to a different environment. [Figure 1-4 on page 1-8](#) shows the paths for migrating between different database servers in the same environment.

Choosing a Migration Method

UNIX and Windows NT store data in different page sizes. INFORMIX-SE stores data in a different format than Universal Server, OnLine Dynamic Server, or OnLine Workgroup Server does. Therefore, when your migration involves different environments or SE, you must export the data and its schema information from one database server and import the exported data into the other database server.

The method that you choose for exporting and importing data depends on how much data you plan to move. All of these methods deliver similar performance and allow you to modify the schema of the database. You can use the following migration methods:

- **dbexport/dbimport**
To move an entire database, use the **dbexport** utility.
- **UNLOAD/dbschema/LOAD**
To move selected columns or tables, use the UNLOAD statement. Use LOAD when you do not wish to change the data format.
- **UNLOAD/dbschema/dbload**
To move selected columns or tables, use the UNLOAD statement. Use **dbload** to change the data format.



- **onunload/onload**

To move data in page-sized chunks, use the **onunload** utility. Use **onunload** to move data to an identical database server on a computer of the same type.

- **High-Performance Loader (HPL)**

To move selected columns or tables or an entire database, use the HPL.

Important: Avoid using **onunload/onload** with Universal Server data; use the HPL instead.

Moving Data from SE to OnLine Dynamic Server or OnLine Workgroup Server

The following sections describe the steps for moving data from SE to OnLine Dynamic Server or OnLine Workgroup Server. You must choose a migration method, eliminate SE-specific features, migrate the data from SE, and modify client applications.

Using the dbexport and dbimport Utilities

If you intend to move an entire database from SE to OnLine Dynamic Server or OnLine Workgroup Server, the **dbexport/dbimport** combination is the easiest migration method.

To use **dbexport** and **dbimport** to move data from SE to OnLine Dynamic Server or OnLine Workgroup Server

1. Use the **dbexport** utility to export the data from SE.

You can move the data to another directory or directly to tape.

Warning: Do not use the **-ss** option with **dbexport** when you move data between database servers. The **-ss** option generates SE-specific syntax that OnLine Dynamic Server or OnLine Workgroup Server does not recognize.



2. Edit the database schema file (the **.sql** file that **dbexport** creates). You might want to add information that OnLine Dynamic Server (or OnLine Workgroup Server) databases and tables can use, such as:
 - Initial- and next-extent values for a table
 - Lock mode for a table
 - Blobspace where TEXT or BYTE data types should reside
 - Dbspace where the tables should reside
 - VARCHAR or NVARCHAR column specifications
 - Fragmentation schemes ♦
3. You can also make the following changes to the schema file:
 - Alter ownership or SQL privileges on specific tables and indexes.
 - Specify the dbspace location of the database.
 - Change the logging status of the database.
 - Change the ANSI-compliance status of the database.
 - Remove unsupported SQL syntax, if necessary. (You can write a script to automate the task of modifying the SQL statements.)

For details about supported SQL statements, see the *Informix Guide to SQL: Syntax*.
4. If necessary, install the new database server. [Figure 8-1 on page 8-8](#) shows where to obtain information on installing and configuring the following database servers.

Figure 8-1
Where to Get Information on Installing Database Servers

Server Type	Manual
ODS/NT	<i>INFORMIX-OnLine Dynamic Server Administration Tools and Database Server Installation Guide</i>
ODS/UNIX	<i>UNIX Products Installation Guide, INFORMIX-OnLine Dynamic Server Administrator's Guide</i>
OWS/NT	<i>INFORMIX-OnLine Workgroup Server Administration Tools and Database Server Installation Guide</i>
OWS/UNIX	<i>INFORMIX-OnLine Workgroup Server Installation Guide for UNIX</i>

5. Change the **INFORMIXSERVER** environment variable to specify your OnLine Dynamic Server (or OnLine Workgroup Server).
6. If necessary, use your standard file-transfer process to move the export files to the target computer. Ensure that both database servers use the same **DBDATE** and **DBMONEY** formats.
7. Start the new database server.
8. Use the **dbimport** utility to move the database data into the new database server.

For detailed descriptions of the **dbexport** and **dbimport** utilities, refer to [Chapter 10, “Utilities for Data Migration.”](#)

Using the UNLOAD Statement

The UNLOAD statement lets you retrieve selected rows from a database and write those rows into a text file.

Using UNLOAD, dbschema, and LOAD

If you want to move selected tables or columns instead of an entire SE database to OnLine Dynamic Server (or OnLine Workgroup Server), use the SQL statements UNLOAD and LOAD in the DB-Access utility with the **dbschema** database utility.

To use UNLOAD, dbschema, and LOAD to move data from SE to OnLine Dynamic Server or OnLine Workgroup Server

1. Make sure that you have sufficient disk space to store the unloaded data. (You cannot unload data to tape with UNLOAD.)
2. Invoke the DB-Access utility.
3. Use the UNLOAD statement to move the selected data into a text file. Use a separate UNLOAD statement for each target table.
4. Exit from DB-Access.
5. If you plan to load data into a table or tables that do not exist, follow these steps:
 - a. Use the **dbschema** utility to create a schema file from the SE database.
 - b. Edit the schema file so that it describes the new tables.
 - c. Make any of the following changes to the schema file:
 - Alter ownership or SQL privileges on specific tables and indexes.
 - Specify the dbspace location of the database.
 - Change the logging status of the database.
 - Change the ANSI-compliance status of the database.
 - Remove unsupported SQL syntax, if necessary.

For details about supported SQL statements, see the *Informix Guide to SQL: Syntax*.

If you prefer, you can omit this step and, in step 12, enter the statements that create the tables.

6. If necessary, install OnLine Dynamic Server (or OnLine Workgroup Server). Figure 8-1 shows where to obtain information on installing and configuring these database servers.
7. Change the **INFORMIXSERVER** environment variable to specify your OnLine Dynamic Server (or OnLine Workgroup Server).
8. If necessary, use your standard file-transfer process to move the export files to the target computer. Ensure that both database servers use the same **DBDATE** and **DBMONEY** formats.
9. Start the new database server.
10. Invoke the DB-Access utility.

11. Select the target database.
If you are creating a new database, execute the CREATE DATABASE statement, or choose **Database→Create** from the DB-Access menu.
12. If you plan to load data into a new table or tables, choose and run the schema file that you prepared in step 5, or enter CREATE TABLE statements to create the new tables.
13. If you created a new database, you might need to update the **DBPATH** environment variable with the new database location.
14. Use LOAD statements to load the data into the desired tables.

Using UNLOAD, dbschema, and dbload

If you need to manipulate the data before you load it into a new table, use a combination of the UNLOAD statement and the **dbschema** and **dbload** utilities.

To use UNLOAD, dbschema, and dbload to move data from SE to OnLine Dynamic Server or OnLine Workgroup Server

1. Follow steps 1 through 13 of [“Using UNLOAD, dbschema, and LOAD” on page 8-8](#).
2. Build a command file to use with the **dbload** utility, which loads the data files into the tables.
3. Execute **dbload** to load the data as your command file directs.

For information on UNLOAD, LOAD, **dbload**, and **dbschema**, refer to [Chapter 10, “Utilities for Data Migration.”](#) For information on using DB-Access, refer to the *DB-Access User Manual*.

Adapting Your Programs for OnLine Dynamic Server or OnLine Workgroup Server

After you successfully move the SE data to OnLine Dynamic Server or OnLine Workgroup Server, verify that your application developers know the differences between both database servers. For more information about the server-specific limitations of SQL statements, refer to the *Informix Guide to SQL: Syntax* and Chapter 1 of the *Informix Guide to SQL: Reference*. For information about environment variables, see Chapter 4 of the *Informix Guide to SQL: Reference*.

ODS

For more information about configuring OnLine Dynamic Server, refer to the *INFORMIX-OnLine Dynamic Server Administrator's Guide*. ♦

OWS

For more information about configuring OnLine Workgroup Server, refer to the *INFORMIX-OnLine Workgroup Server Administrator's Guide*. ♦

SE

The following statements contain syntax and keywords that only SE recognizes:

- CHECK TABLE
- CREATE AUDIT
- DROP AUDIT
- RECOVER TABLE
- REPAIR TABLE
- ROLLFORWARD DATABASE
- START DATABASE ♦

Moving Data from OnLine Dynamic Server or OnLine Workgroup Server to SE

When you move data from OnLine Dynamic Server or OnLine Workgroup Server to SE, you must choose a migration method, eliminate features that SE does not support, migrate the data, and modify applications.

Eliminating Features That SE Does Not Support

Before you export OnLine Dynamic Server (or OnLine Workgroup Server) data to SE, you must eliminate or convert the following unsupported data types to a data type that SE supports:

- VARCHAR
- BYTE
- TEXT

Using the dbexport and dbimport Utilities

If you intend to move an entire database from OnLine Dynamic Server (or OnLine Workgroup Server) to SE, the **dbexport/dbimport** combination is the easiest migration method.

To use **dbexport** and **dbimport** to move data from OnLine Dynamic Server or OnLine Workgroup Server to SE

1. Use **dbexport** to export the data from OnLine Dynamic Server (or OnLine Workgroup Server).

You can move the data to a directory or directly to tape.

Warning: Do not use the **-ss** option with **dbexport** when you move data between database servers. The **-ss** option generates OnLine Dynamic Server or OnLine Workgroup Server-specific syntax that SE does not recognize.



2. Remove the following information from the CREATE TABLE statements in the schema file (the **.sql** file that **dbexport** creates):
 - Initial- and next-extent sizes
 - Lock modes
 - Dbspace names
 - Blobspace names
 - Logging modes
 - Table-fragmentation expressions
3. You can also make the following changes in the schema file:
 - Alter ownership or SQL privileges on specific tables and indexes.
 - Change the logging status of the database.
 - Change the ANSI-compliance status of the database.
4. If necessary, install SE. For installation instructions, refer to the *UNIX Products Installation Guide* and the *INFORMIX-SE Administrator's Guide*.
5. Change the **INFORMIXSERVER** environment variable to specify your new database server.
6. If necessary, use your standard file-transfer process to move the export files to the target computer. Ensure that both database servers use the same **DBDATE** and **DBMONEY** formats.
7. Move to the directory where **dbimport** will store the SE database.
8. Use **dbimport** to move the data into an SE database.

For detailed descriptions of the **dbexport** and **dbimport** utilities, refer to [Chapter 10, “Utilities for Data Migration.”](#)

Using the UNLOAD Statement

The UNLOAD statement lets you retrieve selected rows from a database and write those rows into a text file.

Using UNLOAD, dbschema, and LOAD

If you want to move selected tables or columns instead of an entire OnLine Dynamic Server (or OnLine Workgroup Server) database to SE, use the UNLOAD and LOAD statements in the DB-Access utility with the **dbschema** utility.

To use UNLOAD, **dbschema**, and LOAD to move data from OnLine Dynamic Server (or OnLine Workgroup Server) to SE

1. Make sure that you have sufficient disk space to store the unloaded data. (The UNLOAD statement does not allow you to unload to tape.)
2. Invoke the DB-Access utility.
3. Use UNLOAD statements to move the selected data into text files. Use a separate UNLOAD statement for each target table.
4. Exit from DB-Access.
5. If you plan to load data into a table or tables that do not exist, follow these steps:
 - a. Use the **dbschema** utility to create a schema file from the OnLine Dynamic Server (or OnLine Workgroup Server) database.
 - b. Edit the schema file so that it describes the new tables.If you prefer, you can omit this step and, in step 11, enter the statements that create the tables.
6. If necessary, install SE. For installation instructions, refer to the *UNIX Products Installation Guide* and the *INFORMIX-SE Administrator's Guide*.
7. Change the **INFORMIXSERVER** environment variable to specify your new database server.
8. If necessary, use your standard file-transfer process to move the export files to the target computer. Ensure that both database servers use the same **DBDATE** and **DBMONEY** formats.
9. Invoke the DB-Access utility.

10. Select the target database.
If you are creating a new database, execute the CREATE DATABASE statement, or choose **Database→Create** from the DB-Access menu.
11. If you plan to load data into a new table, choose and run the schema file that you prepared in step 5, or enter CREATE TABLE statements to create the new tables.
12. If you created a new database, you might need to update the **DBPATH** environment variable with the new database location.
13. Use LOAD statements to load the data into the desired tables.

Using UNLOAD, dbschema, and dbload

If you need to manipulate the data before you load it into a new table, use a combination of the UNLOAD statement and the **dbschema** and **dbload** utilities.

To use UNLOAD, dbschema, and dbload to move from OnLine Dynamic Server or OnLine Workgroup Server to SE

1. Follow steps 1 through 12 of [“Using UNLOAD, dbschema, and LOAD” on page 8-14](#).
2. Build a command file to use with the **dbload** utility, which loads the data files into the tables.
3. Execute **dbload** to load the data as your command file directs.

For information on UNLOAD, LOAD, **dbload**, and **dbschema**, refer to [Chapter 10, “Utilities for Data Migration.”](#) For information on how to use DB-Access, refer to the *DB-Access User Manual*.

SE

Adapting Your Programs for SE

After you migrate the OnLine Dynamic Server (or OnLine Workgroup Server) data to SE, verify that your application developers know the differences between both database servers.

SE does not use the ONCONFIG configuration file. SE supports a subset of the environment variables that OnLine Dynamic Server and OnLine Workgroup Server support. For more information about how to use SE, refer to the *INFORMIX-SE Administrator's Guide*. ♦

For more information about the differences between database servers and their interpretation of SQL, refer to Chapter 1 of the *Informix Guide to SQL: Reference*.

Only OnLine Dynamic Server supports the following SQL statements:

- ALTER FRAGMENT
- GRANT FRAGMENT
- REVOKE FRAGMENT
- SET DATASKIP
- SET PDQPRIORITY

Only OnLine Dynamic Server and OnLine Workgroup Server support the following SQL statements:

- CREATE ROLE
- DROP ROLE
- RENAME DATABASE
- SET ISOLATION
- SET LOG
- SET ROLE
- SET SESSION AUTHORIZATION

If you change the name or pathname of the database server, update the **DBPATH** environment variable with the location of the new database. For information about **DBPATH**, refer to the *Informix Guide to SQL: Reference*.

Moving Data from OnLine Dynamic Server to OnLine Workgroup Server

The following sections describe the steps for moving data from OnLine Dynamic Server to OnLine Workgroup Server in the same environment or different environments.

Moving Data in the Same Environment

Migration between OnLine Dynamic Server and OnLine Workgroup Server is automatic if they share the same environment, such as Windows NT. Automatic migration means you do not need to use a migration utility such as **dbexport** to move the data.

Complete all the migration steps except steps 4 and 10, described on pages 8-17 through 8-22.

Moving Data Between Different Environments

When you move data between UNIX and Windows NT, you must choose a migration utility, eliminate database server- and environment-specific features, migrate the data, and modify applications. Complete the following migration steps:

1. Save a copy of the current configuration files.
2. Verify the integrity of the data.
3. Make a final complete backup of OnLine Dynamic Server.
4. If you are migrating to a different environment, export the OnLine Dynamic Server databases.
5. Bring OnLine Dynamic Server off-line.
6. Install and configure OnLine Workgroup Server.
7. Verify port numbers and **services** file.
8. Customize the database server environment. (This step is optional.)
9. Bring OnLine Workgroup Server on-line.
10. Import the databases to OnLine Workgroup Server.

- 11. Verify the integrity of the data.
- 12. Make an initial, complete backup of OnLine Workgroup Server.
- 13. Run UPDATE STATISTICS.

Save a Copy of the Current Configuration Files

Save a copy of the current ONCONFIG files, located in the **etc** subdirectory of your installation directory, and the **sqlhosts** information.

Save the following files, located in various subdirectories of your installation directory (this step is optional):

- **adtcfg**, located in the **aaodir** subdirectory
- **adtmask.***, located in the **dbssodir** subdirectory

Verify the Integrity of the Data

Use the **oncheck** utility to verify the integrity of the data. Figure 8-2 lists the commands for verifying data integrity.

Figure 8-2
Commands for Verifying the Data Integrity

Action	oncheck Command
Check reserve pages	oncheck -cr
Check extents	oncheck -ce
Check system catalog tables	oncheck -cc <i>database_name</i>
Check data	oncheck -cD <i>database_name</i>
Check indexes	oncheck -cI <i>database_name</i>

For information on **oncheck**, refer to the *INFORMIX-OnLine Dynamic Server Administrator's Guide*.

Back Up OnLine Dynamic Server

Use **ontape** or ON-Archive to make a complete (level-0) backup of OnLine Dynamic Server. If you use **ontape**, execute the following command to make a level-0 backup:

```
ontape -s
```

NT

The Windows NT environment does not support ON-Archive. ♦

The tape parameters must specify a valid tape device. Be sure to retain and properly label the tape volume that contains the backup. For more information on making backups, refer to the *INFORMIX-OnLine Dynamic Server Archive and Backup Guide*.

Export the OnLine Dynamic Server Databases

If you are migrating to a different environment (for example, from UNIX to Windows NT), choose one of the following sets of migration utilities:

- **dbexport/dbimport** (see [“Using the dbexport and dbimport Utilities” on page 8-23](#))
- **UNLOAD/dbschema/LOAD** (see [“Using UNLOAD, dbschema, and LOAD” on page 8-24](#))
- **UNLOAD/dbschema/dbload** (see [“Using UNLOAD, dbschema, and dbload” on page 8-25](#))

Skip this step if you are migrating to OnLine Workgroup Server in the same environment.

Bring OnLine Dynamic Server Off-Line

To shut down OnLine Dynamic Server, you can take one of the following actions:

- Use the **onmode -ky** command.
- Use **Services** in the Windows NT **Control Panel**. ♦

NT

To verify that the current OnLine Dynamic Server is off-line, execute the ON-Monitor command and then choose the **Status** option.

NT

UNIX

Install and Configure OnLine Workgroup Server

If you have not already done so, follow the instructions in the *INFORMIX-OnLine Workgroup Server Administration Tools and Database Server Installation Guide* to install and configure OnLine Workgroup Server.

You can install OnLine Workgroup Server and the administration tools on either the same or different computers. ♦

You install the administration tools on a personal computer that runs Windows 95, and the database server on a UNIX computer. The installation program also starts the server agent, which is the communication link between OnLine Workgroup Server and the client on Windows 95. ♦

Use the **Setup** program to specify the network protocol and the computer on which OnLine Workgroup Server looks for the database server definitions (**sqlhosts** and **osahosts** definitions).

Verify Port Numbers and Services File

The **services** file contains service names, port numbers, and protocol information. If you have installed the database server and the administration tools on different computers, verify that the port number listed in the **services** file is the same on the client and on the server computers.

The **services** file resides in the `\windir\system32\drivers\etc` directory. ♦

The **services** file resides in the `/etc/services` directory on the server and in the `\windir\services` directory on the Windows 95 client. ♦

NT

UNIX

Customize the Database Server Environment

If you are an advanced user, you can customize the ONCONFIG configuration file and environment variables for OnLine Workgroup Server. Use a text editor to edit the ONCONFIG file. For more information on configuration parameters, refer to *INFORMIX-OnLine Workgroup Server Administrator's Guide*.

Important: Use the same values for `ROOTOFFSET`, `ROOTSIZE`, and `ROOTPATH` that you used for your old database server.



Use the **setnet32** tool to customize the environment variables on the client. For more information on environment variables, refer to the *Informix Guide to SQL: Reference*.

Bring OnLine Workgroup Server On-Line

The installation program brings OnLine Workgroup Server on-line automatically.

If you customized the database server environment, bring down and restart OnLine Workgroup Server with the Command Center. When you restart OnLine Workgroup Server, the changes to the configuration parameters and environment variables take effect.

NT

To start OnLine Workgroup Server using the Command Center

1. In the **Informix Administration Tools** program group, double-click the **Command Center** icon.
2. In the Command Center, select the database server in the **All Servers** tree view or the **Servers** list box.
3. Choose **Server→On-line**.

For more information, refer to *Using the INFORMIX-Command Center* or General help. ♦

UNIX

To start OnLine Workgroup Server at the UNIX command line, enter `oninit` on the server computer.

For more information, refer to *Using the OnLine Workgroup Server* or OWS Help. ♦

Import the Databases to OnLine Workgroup Server

Use **dbimport**, **LOAD**, or **dbload** to load the databases into OnLine Workgroup Server, depending on which utility you used to export the databases.

NT

UNIX

Verify the Integrity of the Data

Use the **oncheck** utility to verify the integrity of data, as described in [“Verify the Integrity of the Data” on page 8-18](#).

Back Up OnLine Workgroup Server

Use the Backup and Restore tool to make a complete, whole-system backup of OnLine Workgroup Server. For more information on backing up, refer to *Using the INFORMIX-Command Center* or the on-line help.

Double-click the **Backup and Restore** icon in the **Informix Administration Tools** program group. ♦

To launch Backup and Restore, enter **\$INFORMIXDIR/bin/bar** on the UNIX computer where OnLine Workgroup Server is installed. ♦

Run Update Statistics

After you complete the migration procedure, run the UPDATE STATISTICS statement on the server. The UPDATE STATISTICS statement updates the information that OnLine Workgroup Server uses to plan efficient queries. For more information about UPDATE STATISTICS, refer to the *Informix Guide to SQL: Syntax*.

Migration Complete

The first time OnLine Workgroup Server is brought on-line, the **sysmaster** database is built. Check the message log to ensure that the **sysmaster** database build has completed before you allow users to access the database server. After you ensure that client users can access data on OnLine Workgroup Server, the migration process is complete.

Adapting Your Programs for OnLine Workgroup Server

After you successfully move the OnLine Dynamic Server data to OnLine Workgroup Server, verify that your application developers know the differences between both database servers. OnLine Workgroup Server supports the same features as OnLine Dynamic Server, except the following features:

- Fragmentation (also known as partitioning)
- High-Performance Loader (HPL)
- Parallel data query (PDQ)
- Role separation

For information on the SQL statements that OnLine Workgroup Server supports, refer to the *Informix Guide to SQL: Syntax* manual and Chapter 1 of the *Informix Guide to SQL: Reference*.

Using the dbexport and dbimport Utilities

If you intend to move an entire database from OnLine Dynamic Server to OnLine Workgroup Server in different environments, the **dbexport/dbimport** combination is the easiest migration method:

1. Use **dbexport** to export the data from OnLine Dynamic Server.
You can move the data to a directory or directly to tape. Do not use the **-ss** option when you move data between database servers.
2. Remove table-fragmentation expressions from the CREATE TABLE statements in the schema file (the **.sql** file that **dbexport** creates).
If tables are fragmented, **dbimport** might not work. Informix recommends that you defragment the tables before you export the database. ♦
3. You can also make the following changes in the schema file:
 - Alter ownership or SQL privileges for specific tables and indexes.
 - Change the logging status of the database.
 - Change the ANSI-compliance status of the database.
 - Remove unsupported SQL syntax, if necessary.

4. Follow the instructions in the *INFORMIX-OnLine Workgroup Server Administration Tools and Database Server Installation Guide* to install and configure OnLine Workgroup Server.
5. Change the **INFORMIXSERVER** environment variable to specify your new database server.
6. If necessary, use your standard file-transfer process to move the export files to the target computer. Ensure that both database servers use the same **DBDATE** and **DBMONEY** formats.
7. Move to the directory where **dbimport** will store the OnLine Workgroup Server database.
8. Use **dbimport** to move the data to an OnLine Workgroup Server database.

For detailed descriptions of the **dbexport** and **dbimport** utilities, refer to [Chapter 10, “Utilities for Data Migration.”](#)

Using the UNLOAD Statement

The UNLOAD statement lets you retrieve selected rows from a database and write those rows to a text file.

Using UNLOAD, dbschema, and LOAD

If you want to move selected tables or columns instead of an entire OnLine Dynamic Server database to OnLine Workgroup Server, use the UNLOAD and LOAD statements in the DB-Access utility with the **dbschema** utility.

To use UNLOAD, dbschema, and LOAD to move data from OnLine Dynamic Server to OnLine Workgroup Server

1. Make sure that you have sufficient disk space to store the unloaded data. (The UNLOAD statement does not allow you to unload to tape.)
2. Invoke the DB-Access utility.
3. Use UNLOAD statements to move the selected data into text files. Use a separate UNLOAD statement for each target table.
4. Exit from DB-Access.

5. If you plan to load data into a table or tables that do not exist, follow these steps:
 - a. Use the **dbschema** utility to create a schema file from the OnLine Dynamic Server database.
 - b. Edit the schema file so that it describes the new tables.
 - a. If you prefer, you can omit this step and, in step 12, enter the statements that create the tables.
6. Follow the instructions in the *INFORMIX-OnLine Workgroup Server Administration Tools and Database Server Installation Guide* to install and configure OnLine Workgroup Server.
7. Change the **INFORMIXSERVER** environment variable to specify your new database server.
8. If necessary, use your standard file-transfer process to move the export files to the target computer. Ensure that both database servers use the same **DBDATE** and **DBMONEY** formats.
9. Invoke the DB-Access utility.
10. Select the target database.
11. If you are creating a new database, execute the **CREATE DATABASE** statement, or choose **Database→Create** from the DB-Access menu.
12. If you plan to load data into a new table, choose and run the schema file that you prepared in step 5, or enter **CREATE TABLE** statements to create the new tables.
13. If you created a new database, you might need to update the **DBPATH** environment variable with the new database location.
14. Use **LOAD** statements to load the data into the desired tables.

Using UNLOAD, dbschema, and dbload

If you need to manipulate the data in the specified **UNLOAD** file before you load it into a new table, use a combination of the **UNLOAD** statement and the **dbschema** and **dbload** utilities.

To use **UNLOAD**, **dbschema**, and **dbload** to move from OnLine Dynamic Server to OnLine Workgroup Server

1. Follow steps 1 through 13 from [“Using UNLOAD, dbschema, and LOAD” on page 8-24](#).
2. Build a command file to use with the **dbload** utility, which loads the data files into the tables.
3. Execute **dbload** to load the data as your command file directs.

For information on **UNLOAD**, **LOAD**, **dbload**, and **dbschema**, refer to [Chapter 10, “Utilities for Data Migration.”](#) For information on how to use DB-Access, refer to the *DB-Access User Manual*.

Moving Data from OnLine Workgroup Server to OnLine Dynamic Server

The following sections describe the steps for moving data from OnLine Workgroup Server to OnLine Dynamic Server in the same environment or different environments.

Moving Data in the Same Environment

Migration between OnLine Workgroup Server and OnLine Dynamic Server is automatic if they share the same environment, such as Windows NT or UNIX. Automatic migration means you do not need to use a migration utility such as **dbexport** to move the data.

Complete the following migration steps, described in detail on [pages 8-28 through 8-32](#):

1. Save a copy of the current configuration files.
2. Verify the integrity of the data.
3. Make a final complete backup of OnLine Workgroup Server.
4. Bring OnLine Workgroup Server off-line.
5. Install OnLine Dynamic Server.



6. Customize the database server environment. (This step is optional.)
7. Bring OnLine Dynamic Server on-line.
8. Verify the integrity of the data.
9. Make an initial, complete backup of OnLine Dynamic Server.
10. Run UPDATE STATISTICS.

Important: If you want to install on Windows NT, select the **Copy all files, but leave configuration alone** option on the **Run Installation Again** page. The installation program automatically copies the new database server files, and saves the configuration and the database data.

*Do not select the **Copy all files and reconfigure the product** option, or your configuration and database information will be lost.*

After you ensure that client users can access data on OnLine Dynamic Server, the migration process is complete.

Moving Data Between Different Environments

When you move data between UNIX and Windows NT, you must choose a migration utility, eliminate database server- and environment-specific features, migrate the data, and modify applications.

Complete the following migration steps:

1. Save a copy of the current configuration files.
2. Verify the integrity of the data.
3. Make a final complete backup of OnLine Workgroup Server.
4. If you are migrating to a different environment, export the OnLine Workgroup Server databases.
5. Bring OnLine Workgroup Server off-line.
6. Install and configure OnLine Dynamic Server and the administration tools.
7. Verify port numbers and **services** file.
8. Customize the database server environment. (This step is optional.)
9. Bring OnLine Dynamic Server on-line.
10. Import the databases into OnLine Dynamic Server.

11. Verify the integrity of the data.
12. Make an initial, complete backup of OnLine Dynamic Server.
13. Run UPDATE STATISTICS.

Save a Copy of the Current Configuration Files

Save a copy of the current ONCONFIG files, located in the **etc** subdirectory of your installation directory, and the **sqlhosts** information:

- %ONCONFIG% ♦
- \$ONCONFIG, **osahosts**, **sqlhosts**, **ttermcap**, **termcap** ♦

Save the following files, located in various subdirectories of your installation directory (this step is optional):

- **adtcfg**, located in the **aaodir** subdirectory
- **adtmasks.***, located in the **dbssodir** subdirectory

Verify the Integrity of the Data

Use the **oncheck** utility to verify the integrity of data, as described in [“Verify the Integrity of the Data” on page 8-18](#).

Back Up OnLine Workgroup Server

Use the Backup and Restore tool to make a complete, whole-system backup of OnLine Workgroup Server. For more information on backing up, refer to *Using the INFORMIX-Command Center* or the on-line help.

Double-click the **Backup and Restore** icon in the **Informix Administration Tools** program group. ♦

To launch Backup and Restore, enter **\$INFORMIXDIR/bin/bar** on the UNIX computer where OnLine Workgroup Server is installed. ♦

NT

UNIX

NT

UNIX

Export the OnLine Workgroup Server Databases

If you are migrating to a different environment (for example, from UNIX to Windows NT), choose one of the following sets of migration utilities:

- **dbexport/dbimport** (see [“Using the dbexport and dbimport Utilities” on page 8-33](#))
- **UNLOAD/dbschema/LOAD** (see [“Using UNLOAD, dbschema, and LOAD” on page 8-34](#))
- **UNLOAD/dbschema/dbload** (see [“Using UNLOAD, dbschema, and dbload” on page 8-35](#))

Skip this step if you are migrating to OnLine Dynamic Server in the same environment.

UNIX

Shut Down OnLine Workgroup Server (UNIX)

Select **Off-Line** from the **Server Mode** list box in the **General** tab of the **Command Center**. ♦

NT

Shut Down OnLine Workgroup Server (Windows NT)

1. In the Command Center, select the database server in the **All Servers** tree view or the **Servers** list box.
2. Choose **Server→Off-line**. ♦

NT

Install and Configure OnLine Dynamic Server and the Administration Tools (Windows NT)

Install and configure OnLine Dynamic Server and the administration tools according to the instructions in the *INFORMIX-OnLine Dynamic Server Administration Tools and Database Server Installation Guide*. You can install OnLine Dynamic Server and the administration tools on either the same or different computers. ♦

UNIX

The installation program automatically starts OnLine Dynamic Server.

Use **Setup** to specify the network protocol and the computer on which OnLine Dynamic Server looks for the database server definitions. ♦

Install OnLine Dynamic Server (UNIX)

Install and configure OnLine Dynamic Server according to the instructions in the *UNIX Products Installation Guide* and the *INFORMIX-OnLine Dynamic Server Administrator's Guide*. You must be user **root** to install the product. When you finish the installation and system reconfiguration, exit as user **root** and log in as user **informix**. ♦

Verify Port Numbers and Services File

The **services** file contains service names, port numbers, and protocol information. If you have installed the database server and the administration tools on different computers, verify that the port number listed in the **services** file is the same on the client and the server computers.

The **services** file resides in the `\windir\system32\drivers\etc` directory. ♦

The **services** file resides in the `/etc/services` directory on the server and in the `\windir\services` directory on the Windows 95 client. ♦

Customize the Database Server Environment

If you are an advanced user, you can customize the ONCONFIG configuration file and environment variables for OnLine Dynamic Server. Use a text editor to edit the ONCONFIG file. For more information on configuration parameters, refer to the *INFORMIX-OnLine Dynamic Server Administrator's Guide*. For more information on environment variables, refer to the *Informix Guide to SQL: Reference*.

Important: Use the same values for `ROOTOFFSET`, `ROOTSIZE`, and `ROOTPATH` that you used for your old database server.



Bring OnLine Dynamic Server On-Line

The installation program brings OnLine Dynamic Server on-line automatically.

If you customized the database server environment, bring down and restart OnLine Dynamic Server with the Command Center. When you restart OnLine Dynamic Server, the changes to the configuration parameters and environment variables take effect.

NT

To start OnLine Dynamic Server on Windows NT

1. In the **Informix Administration Tools** program group, double-click the **Command Center** icon.
2. In the Command Center, select the database server in the **All Servers** tree view or the **Servers** list box.
3. Choose **Server→On-line**.

For more information, refer to *Using the INFORMIX-Command Center* or the General Help. ♦

UNIX

To start OnLine Dynamic Server on UNIX

1. Enter **oninit** at the command-line prompt. (If you wish to initialize the database server, enter **oninit -i**). ♦

Import the Databases into OnLine Dynamic Server

Use **dbimport**, **LOAD**, or **dbload** to load the databases into OnLine Dynamic Server, depending on which utility you used to export the databases.

Verify the Integrity of the Data

Use the **oncheck** utility to verify the integrity of data, as described in [“Verify the Integrity of the Data” on page 8-18](#).

NT

Back Up OnLine Dynamic Server

Use Backup and Restore in the **Command Center** to make a complete, whole-system backup of OnLine Dynamic Server. For more information, see “Backing Up and Restoring Data” in *Using the INFORMIX-Command Center* or General help. ♦

UNIX

Use ON-Bar, ON-Archive, or **ontape** to make a complete, whole-system backup of OnLine Dynamic Server. For more information, refer to the *INFORMIX-OnLine Dynamic Server Archive and Backup Guide* or the *INFORMIX-OnLine Dynamic Server Backup and Restore Guide*. ♦

Run Update Statistics

After you complete the migration procedure, run the UPDATE STATISTICS statement on the server. The UPDATE STATISTICS statement updates the information that OnLine Dynamic Server uses to plan efficient queries. For more information about UPDATE STATISTICS, refer to the *Informix Guide to SQL: Syntax*.

Migration Complete

The first time OnLine Dynamic Server is brought on-line, the **sysmaster** database is built. Check the message log to ensure that the **sysmaster** database build has completed before you allow users to access the database server. After you ensure that client users can access data on OnLine Dynamic Server, the migration process is complete.

Adapting Your Programs for OnLine Dynamic Server

After you successfully move the OnLine Workgroup Server data to OnLine Dynamic Server, verify that your application developers know the differences between both database servers. OnLine Dynamic Server supports the same features as OnLine Workgroup Server, plus the following:

- Fragmentation (also known as partitioning)
- High-Performance Loader (HPL)

- Parallel data query (PDQ)
- Role separation

For information on the SQL statements that OnLine Dynamic Server supports, refer to the *Informix Guide to SQL: Syntax* manual and Chapter 1 of the *Informix Guide to SQL: Reference*.

Using the *dbexport* and *dbimport* Utilities

If you intend to move an entire database from OnLine Workgroup Server to OnLine Dynamic Server in different environments, the **dbexport/dbimport** combination is the easiest migration method:

1. Use **dbexport** to export the data from the old database server.
You can move the data to a directory or directly to tape. Do not use the **-ss** option when you move data between database servers.
2. You might want to add the following information that OnLine Dynamic Server databases and tables can use:
 - Fragmentation schemes
 - PDQ support ♦
3. You can also make the following changes in the schema file:
 - Alter ownership or SQL privileges for specific tables and indexes.
 - Change the logging status of the database.
 - Change the ANSI-compliance status of the database.
 - Remove unsupported SQL syntax, if necessary.
4. Follow the instructions to install and configure OnLine Dynamic Server.
5. Change the **INFORMIXSERVER** environment variable to specify your new database server.
6. If necessary, use your standard file-transfer process to move the export files to the target computer. Ensure that both database servers use the same **DBDATE** and **DBMONEY** formats.
7. Move to the directory where **dbimport** will store the OnLine Dynamic Server database.
8. Use **dbimport** to move the data to an OnLine Dynamic Server database.

For detailed descriptions of the **dbexport** and **dbimport** utilities, refer to [Chapter 10, “Utilities for Data Migration.”](#)

Using the UNLOAD Statement

The UNLOAD statement lets you retrieve selected rows from a database and write those rows to a text file.

Using UNLOAD, dbschema, and LOAD

If you want to move selected tables or columns instead of an entire OnLine Workgroup Server database to OnLine Dynamic Server, use the UNLOAD and LOAD statements in the DB-Access utility with the **dbschema** utility.

To use UNLOAD, dbschema, and LOAD to move data from OnLine Workgroup Server to OnLine Dynamic Server

1. Make sure that you have sufficient disk space to store the unloaded data. (The UNLOAD statement does not allow you to unload to tape.)
2. Invoke the DB-Access utility.
3. Use UNLOAD statements to move the selected data into text files. Use a separate UNLOAD statement for each target table.
4. Exit from DB-Access.
5. If you plan to load data into a table or tables that do not exist, follow these steps:
 - a. Use the **dbschema** utility to create a schema file from the OnLine Dynamic Server database.
 - b. Edit the schema file so that it describes the new tables.

If you prefer, you can omit this step and, in [step 12](#), enter the statements that create the tables.
6. Follow the instructions to install and configure OnLine Dynamic Server.
7. Change the **INFORMIXSERVER** environment variable to specify your new database server.

8. If necessary, use your standard file-transfer process to move the export files to the target computer. Ensure that both database servers use the same **DBDATE** and **DBMONEY** formats.
9. Invoke the DB-Access utility.
10. Select the target database.
11. If you are creating a new database, execute the **CREATE DATABASE** statement, or choose **Database→Create** from the DB-Access menu.
12. If you plan to load data into a new table, choose and run the schema file that you prepared in step 5, or enter **CREATE TABLE** statements to create the new tables.
13. If you created a new database, you might need to update the **DBPATH** environment variable with the new database location.
14. Use **LOAD** statements to load the data into the desired tables.

Using UNLOAD, dbschema, and dbload

If you need to manipulate the data in the specified **UNLOAD** file before you load it into a new table, use a combination of the **UNLOAD** statement and the **dbschema** and **dbload** utilities.

To use UNLOAD, dbschema, and dbload to move from OnLine Workgroup Server to OnLine Dynamic Server

1. Follow steps 1 through 13 from [“Using UNLOAD, dbschema, and LOAD” on page 8-34](#).
2. Build a command file to use with the **dbload** utility, which loads the data files into the tables.
3. Execute **dbload** to load the data as your command file directs.

For information on **UNLOAD**, **LOAD**, **dbload**, and **dbschema**, refer to [Chapter 10, “Utilities for Data Migration.”](#) For information on how to use DB-Access, refer to the *DB-Access User Manual*.

Moving OnLine Dynamic Server Data Between Environments

The following section describes the steps for moving OnLine Dynamic Server Version 7.22 data between UNIX and Windows NT environments.

Procedure for Moving Data Between Environments

Complete the following migration steps:

1. Save a copy of the current configuration files. For detailed steps, see [“Save a Copy of the Current Configuration Files” on page 8-18](#).
2. Use ON-Bar or **ontape** to make a final level-0 backup.
For more information, refer to the *INFORMIX-OnLine Dynamic Server Archive and Backup Guide* or the *INFORMIX-OnLine Dynamic Server Backup and Restore Guide*.
3. Choose one of the following sets of migration utilities to unload the databases:
 - **dbexport/dbimport**
 - **UNLOAD/dbschema/LOAD**
 - **UNLOAD/dbschema/dbload**
4. Bring OnLine Dynamic Server off-line.
5. Install and configure the new version of OnLine Dynamic Server. If migrating to Windows NT, also install the administration tools.
6. Bring OnLine Dynamic Server on-line.
7. Use **dbimport**, **LOAD**, or **dbload** to load the databases into OnLine Dynamic Server, depending on which utility you used to export the databases.
8. Make an initial level-0 backup under OnLine Dynamic Server.
9. Run **UPDATE STATISTICS** to update the information that the server uses to plan efficient queries.

Using the Migration Utilities

Choose one of the following migration utilities:

- If you intend to move an entire database on OnLine Dynamic Server between different environments, the **dbexport/dbimport** combination is the easiest migration method. Follow the steps in [“Using the dbexport and dbimport Utilities” on page 8-33.](#)
- If you want to move selected tables or columns, instead of an entire database, use the UNLOAD and LOAD statements with the **dbschema** utility. Follow the steps in [“Using the UNLOAD Statement” on page 8-34.](#)
- If you need to manipulate the data in the specified UNLOAD file before you load it into a new table, use a combination of the UNLOAD statement and the **dbschema** and **dbload** utilities. Follow the steps in [“Using UNLOAD, dbschema, and dbload” on page 8-35.](#)

Adapting Your Programs for the UNIX or NT Environment

Certain OnLine Dynamic Server configuration parameters and environment variables are environment dependent. For details, see the *INFORMIX-OnLine Dynamic Server Administrator's Guide* and [Appendix A, “Database Server Environment Variables.”](#)

OnLine Dynamic Server for Windows NT, Version 7.22 supports Workgroup Replication and the administration tools GUI. OnLine Dynamic Server for UNIX, Version 7.22, supports Enterprise Replication but not the administration tools GUI.

Migration Complete

The first time OnLine Dynamic Server is brought on-line, the **sysmaster** database is built. Check the message log to ensure that the **sysmaster** database build has completed before you allow users to access the database server. After you ensure that client users can access data on OnLine Dynamic Server, the migration process is complete. Then you might want to seek ways to obtain maximum performance. For details on topics related to performance, refer to the *INFORMIX-OnLine Dynamic Server Performance Guide*.

Moving OnLine Workgroup Server Data Between Environments

The following section describes the steps for moving OnLine Workgroup Server data between UNIX and Windows NT environments.

Procedure for Moving Data Between Environments

Complete the following migration steps:

1. Save a copy of the current configuration files. For detailed steps, see [“Save a Copy of the Current Configuration Files” on page 8-28](#).
2. Use **ontape** or the Backup and Restore tool to make a complete, whole-system backup.

For more information about making backups, refer to *Using the OnLine Workgroup Server* (UNIX) or *Using the INFORMIX-Command Center* (Windows NT).

3. Choose one of the following sets of migration utilities to unload the databases:
 - **dbexport/dbimport**
 - UNLOAD/**dbschema**/LOAD
 - UNLOAD/**dbschema**/**dbload**
4. Bring OnLine Workgroup Server off-line.
5. Install and configure the new version of OnLine Workgroup Server. If migrating to Windows NT, also upgrade the administration tools. The installation program brings OnLine Workgroup Server on-line automatically.
6. Use **dbimport**, **LOAD**, or **dbload** to load the databases into OnLine Workgroup Server, depending on which utility you used to export the databases.
7. Make a complete, whole-system backup of OnLine Workgroup Server.
8. Run UPDATE STATISTICS to update the information that the server uses to plan efficient queries.

Using the Migration Utilities

Choose one of the following migration utilities:

- If you intend to move an entire database on OnLine Workgroup Server between different environments, the **dbexport/dbimport** combination is the easiest migration method. Follow the steps in [“Using the dbexport and dbimport Utilities” on page 8-23.](#)
- If you want to move selected tables or columns, instead of an entire database, use the UNLOAD and LOAD statements with the **dbschema** utility. Follow the steps in [“Using the UNLOAD Statement” on page 8-24.](#)
- If you need to manipulate the data in the specified UNLOAD file before you load it into a new table, use a combination of the UNLOAD statement and the **dbschema** and **dbload** utilities. Follow the steps in [“Using UNLOAD, dbschema, and dbload” on page 8-25.](#)

Adapting Your Programs for the UNIX or NT Environment

UNIX

OnLine Workgroup Server for UNIX is at Version 7.2. It supports GLS and uses Version 7.12 of the INFORMIX-Command Center. ♦

NT

OnLine Workgroup Server for Windows NT is at Version 7.22. It supports GLS, ON-Bar, Workgroup Replication, the Gateway products, and uses Version 7.22 of the Command Center. ♦

Certain OnLine Workgroup Server configuration parameters and environment variables are environment dependent. For details, see the *INFORMIX-OnLine Workgroup Server Administrator's Guide* and [Appendix A, “Database Server Environment Variables.”](#)

Migration Complete

The first time OnLine Workgroup Server is brought on-line, the **sysmaster** database is built. Check the message log to ensure that the **sysmaster** database build has completed before you allow users to access the database server. After you ensure that client users can access data on OnLine Workgroup Server, the migration process is complete.

Changing Locales

Understanding Database-to-Database Server Compatibility	9-3
Selecting the Database Locale	9-4
Finding the Locale of an Asian (ALS) Database	9-6
Finding the Locale of a Pre-6.0 Database	9-6
Finding the Locale of a 6.x, 7.1, or 7.1x Database	9-6
Finding the Available Locales for Version 7.2x or Universal Server Databases	9-7
Environment Variables Used for Language Support	9-8
Native Language Support (NLS)	9-8
Asian Language Support (ALS)	9-10
Global Language Support (GLS)	9-12
Performing Code-Set Conversion	9-13
Changing from an NLS Locale to a GLS Locale	9-14
Migrating from a Version 6.x or 7.1x Product to Version 7.2x	9-14
Changing from an ALS Locale to a GLS Locale	9-17
Migrating from Version 6.x ALS Products to Version 7.2x	9-17
Changing Environment Variables for ALS Version 6.x	9-18
Migrating from Version 5.x ALS Products to Version 7.2x	9-18
Migrating from Version 4.x ALS Products to Version 7.2x	9-19
Migrating from Version 4.x ASCII Japanese Products to Version 7.2x.	9-20
Reverting from a GLS Locale to an NLS Locale	9-20
Reverting to OnLine Dynamic Server Version 6.0, 7.1x, or OnLine Workgroup Server 7.12	9-21
Reverting from SE 7.2x to an Earlier Version	9-21

Reverting from a GLS Locale to an ALS Locale	9-22
Reverting from Version 7.2x to Version 6.x ALS	9-22
Problems Reverting an Original 7.2x Database to 6.x ALS . . .	9-23
Reverting from Version 7.2x to Version 5.x ALS	9-23
Restrictions on Reverting to Version 5.x	9-24
Reverting from Version 7.2x to Version 4.x ALS or ASCII Product .	9-24

This chapter describes the database server compatibility issues when you migrate between locales. Informix products use the *database locale* when they create new databases or read existing databases. The *locale* consists of a number of parameters, including the language, character set, code set, and collation sequence. Informix products derive the locale of a table from the database where that table resides. The [Guide to GLS Functionality](#) describes locales in detail.

This chapter covers the following topics:

- Understanding database-to-database server compatibility
- Selecting the database locale on different versions of Informix products
- Understanding environment variables used for language support
- Converting to Global Language Support (GLS), Native Language Support (NLS), or Asian Language Support (ALS) locales

Understanding Database-to-Database Server Compatibility

Internationalized databases introduce potential compatibility problems. You can encode the character data of an internationalized database in one of many code sets. The character classification rules used to name the database, its tables, and columns are for a particular language. The indexes on NCHAR and NVARCHAR columns are built in the collation order for a particular language. Each language has its own collation order.

Different versions of Informix database servers store the character and collation rules in different places, and they have different forms of these rules.

When you migrate from one version level of an Informix product to another version level, you must modify the locales appropriately.

Selecting the Database Locale

Figure 9-1 shows the locales that different versions of Informix database servers support.

Figure 9-1
Locales That Informix Database Server Versions Support

Database Server Version	Locales Supported
Universal Server	GLS (Informix-supplied)
7.2 through 7.22	GLS (Informix-supplied)
6.x, 7.1, 7.1UD1, 7.12 through 7.14	NLS (operating system vendor-supplied)
4.x, 5.x, 6.x	ALS (operating system vendor-supplied)
All versions	English

A database inherits its locale from the session that creates it. When you set a locale in the environment of the application development tool that issues the CREATE DATABASE statement, the database inherits that locale. Otherwise, the database uses the default locale for the database server that executes the statement.



Important: Users do not need to migrate locales when upgrading to Universal Server because both OnLine Dynamic Server 7.2x and Universal Server support GLS. For more information, refer to the [Guide to GLS Functionality](#).

Informix Version 7.2x products and Universal Server provide the following types of locales:

- **Informix GLS locales.** The Informix Global Language Support (GLS) locales are the same for all operating systems.
- **Locales compatible with Informix operating systems.** Locales that are compatible with Informix operating systems are compatible with locales native to different operating systems. The operating-system locales might have different definitions from one system to another. In pre-7.2 versions of Informix database servers, Native Language Support (NLS) uses operating-system locales.

If you have an ALS or NLS database, you need to decide, before you migrate the databases to Version 7.2x, whether to use the current locale or to convert to an Informix GLS locale. (Your decision is not irreversible. You can also change the locale after you migrate to Version 7.2x.) Consider the following points:

- Upgrading to Version 7.2x with the current operating-system locale requires no special action on your part. However, distributed queries across dissimilar environments might produce incorrect results because of different locale category definitions.
- Changing from an operating-system locale to a GLS locale requires that you unload and then reload your data. When you unload data from a database that uses an English-language locale, the resulting text files are ASCII files. When you unload data from a database that uses a non-English-language locale, the resulting text files might include 8-bit characters and multibyte characters.

Before you decide how to handle your locales, you need to know the locale that the database uses. The locale information depends on the particular database server you currently use, as follows:

- Customized Asian versions
- English versions before Version 6.0
- English versions after Version 6.0

You also need to know what locales are available for your Version 7.2x database server.

Finding the Locale of an Asian (ALS) Database

The Informix Asian Language Support (ALS) products use locales that are designed to work with them. You do not need to change an ALS locale to a GLS locale. When a Version 7.2x database server opens an ALS database, it automatically converts an ALS locale to a GLS locale. For more information, see [“Changing from an ALS Locale to a GLS Locale” on page 9-17](#).

Finding the Locale of a Pre-6.0 Database

Before Version 6.0, only specially customized Informix products could use locales other than English. The standard English versions of pre-6.0 Informix products do not accommodate alternative locales. These products use the default locale of the operating system. In most cases, the locale of the operating system is English.

Version 7.2x products use the U.S. English locale as the default locale. Therefore, you do not need to change the locale of a pre-6.0 database. The Version 7.2x database server assumes the correct locale when it opens and converts the pre-6.0 database.

Finding the Locale of a 6.x, 7.1, or 7.1x Database

Informix Version 6.0 products introduced the concept of locale with the NLS capability. You set the **DBNLS** environment variable to 1 or 2 to enable the NLS capability and the **LANG** environment variable to identify the locale. For more information, see [“Native Language Support \(NLS\)” on page 9-8](#).

The database locale is stored in two rows of the **sysmaster** system catalog table. These rows have **tabid** values of 90 and 91. The row with **tabid** 90 stores the **COLLATION** category of the database locale. The row with **tabid** 91 stores the **CTYPE** category of the database locale.

Use the following statement to find the locale of an OnLine Dynamic Server or OnLine Workgroup Server database:

```
SELECT UNIQUE dbsname, collate FROM sysmaster:systabnames
```

ODS

OWS

To see the locale names of the COLLATION and CTYPE categories of a database locale, select the database in DB-Access, and enter the following command:

```
SELECT tabname, site FROM systables WHERE tabid = 90 or  
tabid = 91
```

In the resulting display, the **dbname** column lists the databases on your database server. The entry in the **collate** column shows the NLS locale of the locale. If the **collate** column is blank, the database is not an NLS database. ♦

To see the locale names of the COLLATION and CTYPE categories for an SE database, select the database in DB-Access, and enter the following command:

```
SELECT dirpath FROM systables WHERE tabid = 90 or tabid = 91;
```

Important: The *glfiles* utility is not available with pre-7.2 database server products.

Finding the Available Locales for Version 7.2x or Universal Server Databases

Before you change the locale, you must also ascertain what locales are available on your database server and on the computers where the client applications reside.

To find the locales that your Version 7.2x or Universal Server databases use, execute the following command in DB-Access:

```
SELECT * FROM sysmaster:sysdbslocale
```

To find out what locales are available, you can run the **glfiles** utility. This utility lists the files that are available on your system. For more information about available locales, contact your local Informix representative.

Important: The *glfiles* utility is only available with products that support GLS.

For information about the **glfiles** utility, refer to Appendix A of the [Guide to GLS Functionality](#). ♦

SE



UNIX



Environment Variables Used for Language Support

Informix Version 7.2x products and Universal Server support many native languages that use both single-byte and multibyte characters. In Version 6.0, Informix Native Language Support (NLS) introduced support for single-byte native languages in all products. Before Version 6.0, support for languages other than English required locally customized versions of Informix products. The following sections provide a brief background of Informix products that support languages other than English.

For more information about specific environment variables, refer to the [Informix Guide to SQL: Reference](#).

Native Language Support (NLS)

Informix introduced NLS for single-byte code sets in its Version 6.0 products. An NLS product uses locales and code sets that the operating system supplies to provide language support. NLS supports only single-byte code sets; it does not support multibyte code sets. Users of multibyte code sets relied on locally customized versions of the products such as the Informix ALS products.

An NLS product uses both operating-system environment variables and Informix NLS environment variables. Figure 9-2 lists Informix NLS environment variables.

Figure 9-2
NLS Environment Variables

NLS Environment Variable	Purpose
COLLCHAR	Allows client applications to use NLS collation
DBAPICODE	Allows client applications to use a different code set than the one that the database server uses
DBDATE	Specifies an end-user format for values in DATE columns
DBLANG	Specifies the location of product-specific message files
DBMONEY	Specifies an end-user format for values in MONEY columns

(1 of 2)

NLS Environment Variable	Purpose
DBNLS	Enables NLS features
DBTIME	Specifies an end-user format for values in DATETIME columns
LANG	Specifies the operating-system locale for your NLS features
LC_COLLATE	Specifies a collating sequence for your locale-sensitive data
LC_CTYPE	Affects the behavior of regular expressions and character-evaluation functions
LC_MONETARY	Specifies the format and national currency symbol for monetary values
LC_NUMERIC	Specifies the format and decimal separator for numeric values
LC_TIME	Specifies the format for national dates and times

(2 of 2)

For more information about how to use NLS products, refer to the *Informix Guide to SQL: Reference*, Version 7.1; the *INFORMIX-ESQL/C Programmer's Manual*, Version 6.0; and the *INFORMIX-ESQL/COBOL Programmer's Manual*, Version 6.0.

Version 7.2x Support of NLS Environment Variables

Informix Version 7.2x products support all NLS environment variables. Therefore, you can continue to use an NLS client application with a Version 7.2x database server or Version 7.2x client applications with an NLS database server with minimal modification. For more information, see [“Changing from an NLS Locale to a GLS Locale” on page 9-14](#). For more information on GLS environment variables, see [“Global Language Support \(GLS\)” on page 9-12](#).

Asian Language Support (ALS)

Before Version 7.2x, locally customized versions of Informix products supported Asian languages that used multibyte characters. Version 4.x ASCII Japanese (4.s) and Informix ALS products Versions 4.x, 5.x, and 6.x support multibyte characters.

An ALS product uses locales that Informix provides for Asian-language support. Figure 9-3 lists Informix ALS environment variables.

Figure 9-3
ALS Environment Variables

ALS Environment Variable	Purpose	Supported by Version 7.2
ALS8BITFSYS	Enables ALS products to handle 8-bit filenames	No (use GLS8BITFSYS instead)
CC8BITLEVEL	Specifies the preprocessing format for the ESQL/C compiler	Yes
CLIENT_LOCALE	For Version 5.x ALS, specifies the code set that the client application uses. For Version 6.x ALS, specifies the locale that the client application uses.	Yes
DBCODESET	Similar to CLIENT_LOCALE and DB_LOCALE; used by Version 5.x ALS	No
DBCSOEVERIDE	Forces DB_LOCALE values to override default restrictions on accessing databases	No
DBDATE	Specifies an end-user format for values in DATE columns	Yes (use GL_DATE instead)
DBFORMAT	Specifies formats for INFORMIX-SQL and INFORMIX-4GL	No
DBINFXRC	Specifies the pathname of a user-defined environment-variable configuration file	No
DBLANG	Specifies the location of product-specific message files	Yes

(1 of 2)

ALS Environment Variable	Purpose	Supported by Version 7.2
DB_LOCALE	For Version 5.x ALS, specifies the code set that locale-sensitive data in the database uses. For Version 6.x ALS, specifies the locale of the database.	Yes
DBMONEY	Specifies an end-user format for values in MONEY columns	Yes
DBTIME	Specifies an end-user format for values in DATETIME columns	Yes

(2 of 2)

For more information about how to use ALS products, refer to your ALS documentation.

Version 7.2x Support of ALS Environment Variables

Informix Version 7.2x products support most ALS environment variables. Therefore, you can continue to use an ALS client application with a Version 7.2x database server or Version 7.2x client applications with an ALS database server. For more information, see [“Changing from an ALS Locale to a GLS Locale” on page 9-17](#). For more information on GLS environment variables, see [“Global Language Support \(GLS\)” on page 9-12](#).

Global Language Support (GLS)

Informix Version 7.2x products and Universal Server support both single-byte and multibyte characters. Informix supplies GLS locales for the languages that are supported.

Informix Version 7.2x products and Universal Server support all of the NLS environment variables (shown in [Figure 9-2 on page 9-8](#)) and a subset of the ALS environment variables (shown in [Figure 9-3 on page 9-10](#)). GLS supports additional environment variables that relate to locale and language shown in [Figure 9-4](#).

Figure 9-4
GLS Environment Variables

GLS Environment Variable	Purpose
CC8BITLEVEL	Specifies how the C compiler handles multibyte characters
CLIENT_LOCALE	Specifies the client locale
DB_LOCALE	Specifies the database locale
ESQLMF	Specifies whether to invoke the ESQL/C multibyte filter, esqlmf
GL_DATE	Supports extended format strings for international formatting of DATE values
GL_DATETIME	Supports extended format strings for international formatting of DATE values
GLS8BITFSYS	Specifies how to handle filenames that contain non-ASCII characters
SERVER_LOCALE	Specifies the database server locale

Performing Code-Set Conversion

Figure 9-5 shows what the pre-7.2 client products send to a Version 7.2x database server when they establish a connection. The Client Locale and Database Locale columns show how the Version 7.2x database server converts this client information to a client and database locale, respectively. The Code-set Conversion column shows the environment variables that the pre-7.2 client products use to convert the code-set. If the database locale is code-set ordered, the database server does not have to convert any system table columns.

Figure 9-5
Locale Information Sent to Version 7.2x Database Servers

Client Product	Version 7.2x Database Server		
	Client Locale	Database Locale	Code-Set Conversion
7.2 ESQL/C, ESQL/COBOL 5.x ESQL/C for Windows	CLIENT_LOCALE	DB_LOCALE	CLIENT_LOCALE <-> DB_LOCALE
pre-7.2, 6.x NLS ESQL/C, ESQL/COBOL	LANG, LC_* (with DBNLS=1)	LANG, LC_* (with DBNLS=1)	DBAPICODE
6.x ALS ESQL/C	CLIENT_LOCALE	DB_LOCALE	CLIENT_LOCALE <-> DB_LOCALE
7.1 ESQL/C for Win32 5.x ESQL/C for Windows	LANG, LC_* (with DBNLS=1)	LANG, LC_* (with DBNLS=1)	CLIENT_LOCALE <-> DB_LOCALE
5.x ALS ESQL/C	DBCODESET	DBCODESET	DBAPICODE <-> DBCODESET

For more information on GLS, refer to the [Guide to GLS Functionality](#).

Changing from an NLS Locale to a GLS Locale

This section discusses how to change from an NLS locale to a GLS locale when you migrate from a Version 6.x or 7.x database server to Version 7.2x. Before you migrate to Version 7.2x, you must modify the locales appropriately.

The Version 7.2x database server stores a condensed version of the database locale in the **systables** system catalog table. The rows with the values 90 and 91 in the **tabid** column of the **systables** system catalog table store the locale name in the **site** column. These **systables** rows contain information that is used throughout the lifetime of the database for such operations as handling regular expressions, collating character strings, and ensuring proper use of code sets. At runtime, the database server maps this database locale to a locale file on the system where the database resides.

The **tablename** values of rows 90 and 91 in the **systables** system catalog table are different in the pre-7.2 and 7.2x versions. In Version 7.2x, the **tablename** value for the row whose **tabid** is 90 is GL_COLLATE. In Version 7.2x, the **tablename** value for the row whose **tabid** is 91 is GL_CTYPE.

Migrating from a Version 6.x or 7.1x Product to Version 7.2x

NLS databases use locales that the operating system provides and single-byte code sets. GLS databases use locales that Informix provides. Before you move a database to Version 7.2x, you must determine whether your database uses an NLS locale, and if so whether the Version 7.2x product provides an equivalent GLS locale.



Important: Check the availability of the GLS locale before you open NLS databases with a Version 7.2x database server.

To determine the NLS locale of your database

1. Set the appropriate NLS environment variables (see [Figure 9-2 on page 9-8](#)).
2. Start a pre-7.2 database server.
3. Enter DB-Access.
4. Open your database.
5. Use the DB-Access Query option to choose one of the following commands, depending on your database server type:

```
SELECT tabname, site FROM systables
WHERE tabid = 90 OR tabid = 91
```

The value displayed in the **site** column is the name of your current NLS locale. ♦

```
SELECT dirpath FROM systables WHERE tabid = 90
OR tabid = 91;
```

The value in the **dirpath** column is the name of your current NLS locale. ♦

If no matching entry is found in **systables**, your database is not an NLS database; that is, the database uses the default English-language locale.

To determine whether a compatible GLS locale exists for your current (NLS) locale

1. Set your \$INFORMIXDIR and \$PATH environment variables to point to a Version 7.2x database server.
2. Execute the **glfiles** utility.
The **glfiles** utility examines the information that is supplied with your installation. It lists the locales that are available in a file called **lcX.txt**, where **X** is the version of the locale files. For more information on **glfiles**, refer to the [Guide to GLS Functionality](#).
3. Examine the **lcX.txt** file for a GLS locale name that is compatible with your NLS locale. If you do not find an operating-system-compatible locale file for your NLS locale, try to find the closest fit.
4. Update the **systables** rows 90 and 91 with this GLS locale name. ♦

ODS

SE

UNIX

To open the pre-7.2 NLS database with the Version 7.2x database server

Once you have ensured that the system catalog table contains a GLS locale, you can open an NLS database with a Version 7.2x database server. Otherwise, you cannot open the NLS database with a Version 7.2x database server.

1. Set the GLS environment variables, `CLIENT_LOCALE` and `DB_LOCALE`, to the appropriate locale for the NLS database. (For more information on GLS environment variables, see [Figure 9-4 on page 9-12.](#))
2. When the Version 7.2x database server opens an NLS database, it changes the **systables** system catalog table, as follows:
 - Changes the **tabname** value from `NLSCOLL` to `GL_COLLATE` in the row whose **tabid** value is 90.
 - Changes the **tabname** value from `NLSCTYPE` to `GL_CTYPE` in the row whose **tabid** value is 91.
 - Creates a condensed locale name from the locale name in rows 90 and 91 of **systables**. If a locale corresponding to the condensed name exists, the database server stores the condensed locale name in rows 90 and 91. If the existing locale does not have a condensed locale name, the database server does not change the locale name in rows 90 and 91.
3. If the locale defines a localized order for collation, the database server converts the data type of all `CHAR` and `VARCHAR` columns of the system catalog table to `NCHAR` and `NVARCHAR`.

The default locale, U.S. English, does not define a localized order. If the locale does not define a localized order, the database server uses the code-set order for collation of all characters. It does not convert `CHAR` and `VARCHAR` columns of the system catalog table.

To open a non-NLS (English) database with the Version 7.2x database server

You do not need to set the GLS environment variables before you open a non-NLS database with a Version 7.2x database server. Version 7.2x products use the default U.S. English locale automatically. When you open a non-NLS database that was created prior to Version 7.2x, the Version 7.2x database server changes the following in the **systables** system catalog table:

- Creates a row with the **tablename** of GL_COLLATE. The **tabid** of the row is 90.
- Creates a row with the **tablename** of GL_CTYPE. The **tabid** of the row is 91.
- Stores a condensed locale name for the default locale, **en_us.819** in rows 90 and 91.

Changing from an ALS Locale to a GLS Locale

This section discusses how to change from an ALS locale to a GLS locale when you migrate from Version 5.x and 6.x ALS database servers to Version 7.2x database servers. When you open an ALS database with a Version 7.2x database server, it copies ALS locale information into rows 90 and 91 of the **systables** system catalog table.

Migrating from Version 6.x ALS Products to Version 7.2x

Informix Version 6.x ALS databases store locale information in the **systables** system catalog table in the rows whose **tabid** values are 95 and 96. When you use a Version 7.2x database server to open a Version 6.x ALS database, the database server takes the following actions:

1. Changes the **systables** system catalog table, as follows:
 - Copies the data in the rows whose **tabid** values are 95 and 96 to rows 90 and 91, respectively.
 - Assigns the **tablename** GL_COLLATE to row 90.
 - Assigns the **tablename** GL_CTYPE to row 91.
2. Changes the locale name to the equivalent Version 7.2x format.

Changing Environment Variables for ALS Version 6.x

Before you migrate from an ALS database to Version 7.2x, set the **GLS8BITFSYS** environment variable to the same value as **ALS8BITFSYS**.

Migrating from Version 5.x ALS Products to Version 7.2x

Version 5.x ALS databases store code-set aliases in the database. When you use a Version 7.2x database server to open a Version 5.x ALS database, the database server performs the following actions:

- 1. Maps the 5.x code-set aliases to a locale name with the table shown in Figure 9-6. If the code set stored in the 5.x ALS database is not one of these aliases, the conversion fails. The conversion also fails if the code set has been customized.
- 2. Stores the mapped GLS locale name in **systables** in the rows whose **tabid** values are 90 and 91.

Version 5.x ALS Code-Set Name	GLS Locale Name
big5	zh_TW.big5
sbig5	zh_TW.sbig5
ccdc	zh_TW.ccdc
ccci	zh_TW.ccci
8859-1	en_us.8859-1
gb2312	zh_CN.gb
ks5601	ko_KR.ks
jis208	ja_jp.ujis
sjis208	ja_jp.sjis

Figure 9-6
*Locale-Name
Conversion for a
Version 5.x ALS
Database*



Warning: This restriction applies to 5.x ALS databases that use a **DBCODESET** of **sjis** and user-defined double-byte characters that are mapped between **f040** and **fcfc**. Do not open the 5.x ALS database with a Version 7.2x database server. Instead, use the **dbexport/dbimport** utilities or **UNLOAD/LOAD** commands in **DB-Access** to migrate the 5.x ALS database to Version 7.2x.

*Do not issue the CREATE DATABASE SQL statement from 5.x ALS client applications that use a DBCODESET of **sjis**. Instead, use DB-Access, Version 7.2x, to create new databases. However, you can use 5.x ALS client applications to create new tables, indexes, views, and so forth.*

Migrating from Version 4.x ALS Products to Version 7.2x

When you use a Version 7.2x database server to open a Version 4.x ALS database, the database server performs the following actions:

1. Maps the 4.x locale that is stored in row 98 of the **systables** system catalog table to a GLS locale name, as Figure 9-7 shows. If the code set that is stored in the 4.x ALS database is not one of these aliases, the conversion fails.
2. Stores the mapped locale name in **systables** in the **site** column of the rows whose **tabid** values are 90 and 91.

4.1 ALS Code-Set Name	GLS Locale Name
ascii	en_us.8859-1
gb	zh_CN.gb
ksc	ko_KR.ksc
ksc1	ko_KR.ksc
big5	zh_TW.big5
sbig5	zh_TW.sbig5
ccdc	zh_TW.ccdc
ccci	zh_TW.ccci

Figure 9-7
Locale-Name
Conversion for a
Version 4.1 ALS
Database



Warning: This restriction applies to 4.x ALS databases that use a DBCODESET of **sjis** and user-defined double-byte characters that are mapped between f040 and fcfc. Do not open this 4.x ALS database with a Version 7.2x database server. Instead, use the **dbexport/dbimport** utilities or UNLOAD/LOAD commands in DB-Access to migrate the 4.x ALS database to Version 7.2x.

*Do not issue the CREATE DATABASE SQL statement from 4.x ALS client applications that use a DBCODESET of **sjis**. Instead, use DB-Access, Version 7.2x, to create new databases. However, you can use 4.x ALS client applications to create new tables, indexes, views, and so forth.*

Migrating from Version 4.x ASCII Japanese Products to Version 7.2x

The GLS database server cannot convert 4.x ASCII databases (Japanese-language version of 4.x ALS) to GLS databases. You must use the **dbexport/dbimport** utilities or the UNLOAD/LOAD commands to unload the 4.x ASCII database and then load it into a Version 7.2x database server.



Warning: *Do not issue the CREATE DATABASE SQL statement from 4.x ASCII client applications. Instead, use DB-Access, Version 7.2x, to create new databases. However, you can use 4.x ASCII client applications to create new tables, indexes, views, and so forth.*

Reverting from a GLS Locale to an NLS Locale

The GLS database reverts to an NLS database when you revert from Version 7.2x to a pre-7.2 version of OnLine Dynamic Server or OnLine Workgroup Server.

The database server automatically moves the locale entries in **systables** during the reversion process. For a Version 6.x or 7.x NLS database server, you can open the database if the operating system accepts the locale that is stored in the database.



Important: *Before you revert to an earlier version, verify that the locales that your Version 7.2x databases use are available as operating-system locales on your computer. Also verify that NLS products support these locales.*

Reverting to OnLine Dynamic Server Version 6.0, 7.1x, or OnLine Workgroup Server 7.12

OnLine Dynamic Server 6.0, 7.1, 7.1UD1, and 7.1x, and OnLine Workgroup Server 7.12 support NLS locales only.

To revert from a Version 7.2x database to an NLS locale

1. Use the following command, where *reversion_level* is one of the values shown in Figure 9-8:

```
onmode -b reversion_level
```
2. Set the appropriate NLS environment variables (described in [Figure 9-2 on page 9-8](#)). NLS databases ignore the CLIENT_LOCALE, DB_LOCALE, and SERVER_LOCALE environment variables.

Target Version	<i>reversion_level</i> Value
6.0	6.0
7.1UC1	7.1
7.1UD1, 7.11, 7.12, 7.13, or 7.14	7.1UD1

Figure 9-8
reversion_level
Values for NLS
Locales

If the database has a locale of **en_us.8559-1**, and no user-defined tables with NCHAR or NVARCHAR columns, the database reverts to a non-NLS database. The **onmode** utility deletes the locale entries in **systables**, converts NCHAR columns in the system catalog table to CHAR columns, and remakes the indexes in the system catalog tables.

If the database has any nondefault locale, or if any user-defined table has NCHAR or NVARCHAR columns, the database reverts to an NLS database. The locale entries in **systables** change to the NLS format (**tablename** from rows 90 and 91 changes from GL_COLLATE and GL_CTYPE to NLSCOLL and NLSCTYPE, respectively). You do not need to remake the indexes.

Reverting from SE 7.2x to an Earlier Version

INFORMIX-SE does not provide a direct path for reverting from Version 7.2x to earlier versions. For more information, refer to [Chapter 7, “Migrating SE.”](#)



Reverting from a GLS Locale to an ALS Locale

This section discusses how to change a GLS locale to an ALS locale when you revert to an Informix ALS product. Informix ALS databases of Versions 4.x, 5.x, and 6.x support ALS locales only. ALS locales support multibyte code sets.

The database server automatically moves the locale entries in **systables** during the reversion process. You can open the database if the database server that you use understands that locale.

Warning: Before you convert to an ALS product, check that it supports the locale that the GLS database uses.

Reverting from Version 7.2x to Version 6.x ALS

To revert from a Version 7.2x database to Version 6.x ALS, follow these steps:

1. Verify that the current GLS locale is supported by the Version 6.x ALS product.
2. Either replace NCHAR columns with CHAR columns and NVARCHAR columns with VARCHAR columns, or drop the NCHAR or NVARCHAR columns from your database.
3. To revert, execute the following command:

```
onmode -b 6.0A
```
4. Set the appropriate ALS environment variables. For more information, see [Figure 9-3 on page 9-10](#).

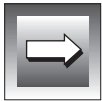
During reversion, the **onmode** command moves the locale entries in **systables** to the 6.0 ALS database location (from rows 90 and 91 to rows 95 and 96), converts NCHAR columns in the system catalog table to CHAR columns, and remakes the indexes in the system catalog table.

Warning: If the database has NCHAR or NVARCHAR in any user-defined table, **onmode** fails.



Problems Reverting an Original 7.2x Database to 6.x ALS

Suppose you convert a database from 6.x ALS to Version 7.2x, then revert it back to 6.x ALS. This type of reversion does not pose a problem because the original locale entries in rows 95 and 96 have not been removed. However, when you revert a database *created* in Version 7.2x, the **onmode** utility might encounter some problems. The **onmode** utility moves the locale entries from rows 90 and 91 to rows 95 and 96. It converts the NCHAR columns of the system catalog table to CHAR, and rebuilds the indexes on these columns. However, the 6.x ALS database server does not interpret the GLS locale name even though that GLS locale is supported. For example, both Version 6.x ALS and 7.2x support the locale **en_us.8859-1**. Version 6.x ALS stores this locale as **en_us.8859-1**, but Version 7.2x stores it as **en_us.819**, which 6.x ALS does not understand.



Reverting from Version 7.2x to Version 5.x ALS

Important: *There is no way to revert from a 7.2x database server to a 5.x ALS database server automatically. However, you can revert to 5.x ALS manually, as follows:*

1. Verify that the current GLS locale is supported in Version 5.x ALS. (See [Figure 9-6 on page 9-18.](#))
2. Choose one of the following:
 - Drop all NCHAR and NVARCHAR columns from the database.
This solution means that you would lose the locale-specific data, but would ensure that the 5.x database server would support the remaining data.
 - Convert the NCHAR columns to CHAR and NVARCHAR columns to VARCHAR.
This solution means that the database server would no longer collate the data in a localized order.
3. To revert, execute the following command:


```
onmode -b 5.0
```
4. Set the appropriate ALS environment variables. For more information, see [Figure 9-3 on page 9-10.](#)

All databases on OnLine 5.0 have to be in **en_us.8859-1** or **en_us.819** for both CTYPE and COLLATE.

Restrictions on Reverting to Version 5.x

If you use locale-specific data (NCHAR and NVARCHAR) and then revert to a database server that does not support non-English data, you would lose access to this data. The **onmode** utility fails when it attempts to convert a Version 7.2x database that contains NCHAR or NVARCHAR columns to a Version 5.x database.

Reverting from Version 7.2x to Version 4.x ALS or ASCII Product

Version 7.2x products do not support reversion to ALS or ASCII Version 4.x.

Section III



Utilities for Data Migration

Setting Environment Variables	10-6
The dbexport Utility	10-7
Using the -c Option	10-9
Using the -ss Option	10-9
Destination Options	10-10
The Contents of the Schema File	10-12
Exporting Simple Large Objects	10-12
The dbimport Utility	10-14
Using the -c Option	10-16
Using Delimited Identifiers with dbimport	10-16
Canceling dbimport	10-16
Database-Logging Mode	10-17
Input-File Location Options	10-17
Create Options	10-19
Using the -l Options	10-20
Changing the Database Name.	10-21
Importing Simple Large Objects	10-22
Using dbimport to Change the Locale of a Database	10-22
The dbload Utility	10-23
Using the dbload Utility.	10-25
Using dbload with SE	10-25
Using the -k Option	10-25
Using the -r Option	10-26
Using the -i Option	10-26
Using the -e and -p Options	10-26
Choosing Between dbload and LOAD	10-27
Creating a dbload Command File	10-27
FILE and INSERT Statements: Delimiter Form.	10-28
FILE and INSERT Statements: Character-Position Form	10-33

The dbschema Utility	39
Creating the Schema for a Database	41
Creating Schemas for Databases Across a Network	42
Owner Naming with dbschema	42
Using the -ss Option	43
Using the -u Option	43
Obtaining the Synonym Schema.	45
Obtaining the Privilege Schema	46
Displaying Privilege Information for a Role.	47
Obtaining the Role Schema	48
Displaying Information on Roles	48
Specifying a Table, View, or Procedure	49
Using the -ss Option to Obtain Table Information.	51
Displaying the Distribution Information for Tables	51
Example Output	52
Distribution Description	52
The Distribution Information.	53
The Overflow Information.	54
The onload Utility	56
Specify Source Parameters	57
Create Options	58
Constraints That Affect onload and onunload	59
Constraints That Affect onload	60
Logging While Using onload	60
Moving Simple Large Objects to a BlobSpace.	61
Ownership and Privileges	61
Exclusive Locking During Load Operation	61
The onmode Utility	64
Preparing to Use the -b Option	64
Reverting to OnLine Version 5.0	65
Reverting to OnLine Dynamic Server Version 6.0	66
Reverting to Version 7.2	67
The onunload Utility	70
Syntax	70
Specify Destination Parameters	71
Constraints That Affect onunload	71
Unloading a Database or Table	72
Unloading a Database	72
Unloading a Table	73

Logging Mode.	73
Locking During Unload Operation.	73
The LOAD Statement.	76
The UNLOAD Statement	77
onpload: The High-Performance Loader Utility	78

This chapter contains information about tools that you can use to move data on Universal Server, OnLine Dynamic Server, OnLine Workgroup Server, and INFORMIX-SE.

The chapter contains syntax and use information for the following utilities:

- The **dbexport** utility unloads a database into text files for later import into another database.
- The **dbimport** utility creates and populates a database from text files.
- The **dbload** utility loads data into databases or tables that were created with Informix products.
- The **dbschema** utility creates a file that contains the SQL statements necessary to replicate a specified table, view, or database. It also shows the distributions that UPDATE STATISTICS creates.
- The **onload** utility loads data that was created with the **onunload** command into the database server.
- The **onunload** utility unloads data from an OnLine Dynamic Server or OnLine Workgroup Server database.
- The **onmode** utility modifies the data in an Informix database so that an earlier version of the database server can access it. (Other functions of **onmode** are documented in the administrator's guide for your database server.)

The chapter also contains brief descriptions of the following tools that are described in other manuals:

- The LOAD statement allows you to append rows to an existing table in a database. For complete documentation of the LOAD statement, refer to the [Informix Guide to SQL: Syntax](#).
- The UNLOAD statement lets you write rows extracted by a SELECT statement to a file. For complete documentation of the LOAD statement, refer to the [Informix Guide to SQL: Syntax](#).



SE

Setting Environment Variables

Before you use any of these migration utilities, you must set your **PATH**, **INFORMIXDIR**, and **INFORMIXSERVER** environment variables. For information about environment variables, see the [Informix Guide to SQL: Reference](#).

***Tip:** If you are using INFORMIX-SE, Informix recommends that you not use the **.dbs** directory as your current directory when you use a database-related utility. This practice keeps the **.dbs** directory free from file clutter and prevents multiple users from overwriting files that belong to other users.*

For information about SE administration utilities, see the *INFORMIX-SE Administrator's Guide*. ♦

The dbexport Utility

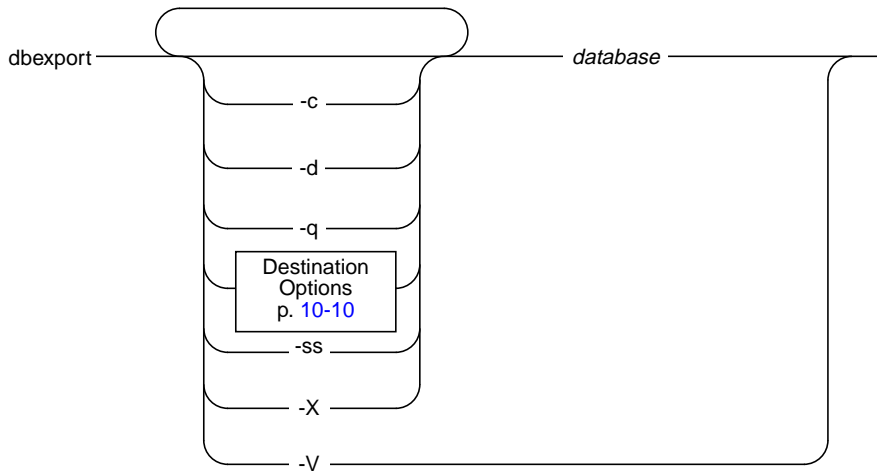
The **dbexport** utility unloads a database into text files and creates a schema file. You can use the schema file with **dbimport** to re-create the database schema in another Informix environment. You can edit the schema file to modify the database that **dbimport** creates. The **dbexport** utility supports Universal Server data types.

The **dbexport** utility supports the following destination options:

- Unload a database and its schema file to disk.
- Unload a database and its schema file to tape.
- Unload the schema file to disk, and unload the data to tape.



***Tip:** The examples in this chapter apply to Universal Server, OnLine Dynamic Server, and OnLine Workgroup Server unless marked with an icon for SE.*



Element	Purpose	Key Considerations
-c	Makes dbexport complete exporting unless a fatal error occurs.	References: For specific details on this option, see “Using the -c Option” on page 10-9 .
-d	Makes dbexport export simple-large-object descriptors only, not simple-large-object data.	References: For more information about simple-large-object descriptors, refer to the <i>INFORMIX-OnLine/Optical User Manual</i> . Restrictions: Not supported by SE.
-q	Suppresses the display of error messages, warnings, and generated SQL data-definition statements.	None.
-ss	Generates database server-specific information for all tables in the specified database.	References: For specific details on this option, see “Using the -ss Option” on page 10-9 .
-X	Recognizes HEX binary data in character fields.	None.
-V	Displays product version information.	None.
database	Specifies the name of the database that you want to export.	Additional Information: If your locale is set to use multibyte characters, you can use multibyte characters for the database name. References: If you want to use more than the simple name of the database, refer to the Database Name section of the Informix Guide to SQL: Syntax .

GLS

You must have DBA privilege or log in as user **informix** to export a database.

When the environment variables are set correctly, as described in the [Guide to GLS Functionality](#), **dbexport** can handle foreign characters in data and export the data from GLS databases. For more information, refer to [“Changing the Database Name” on page 10-21](#). ♦

You can use delimited identifiers with the **dbexport** utility. The utility detects database objects that are keywords, mixed case, or have special characters and places double quotes around them.

In addition to the data files and the schema file, **dbexport** creates a file of messages called **dbexport.out** in the current directory. This file contains error messages, warnings, and a display of the SQL data definition statements that it generates. The same material is also written to the standard output unless you specify the **-q** option.



During the export, the database is locked in exclusive mode. If **dbexport** cannot obtain an exclusive lock, it exits and displays a diagnostic message.

You can press the INTERRUPT key at any time to cancel **dbexport**. The **dbexport** utility asks for confirmation before it terminates.

***Warning:** Use **dbload** instead of **dbexport** if you need to access the database while the database is unloading, because **dbexport** locks the database.*

Using the -c Option

Even if you use the -c option, **dbimport** interrupts processing if one of the following fatal errors occurs:

- Unable to open the tape device specified
- Bad writes to the tape or disk
- Invalid command parameters
- Cannot open database or no system permission

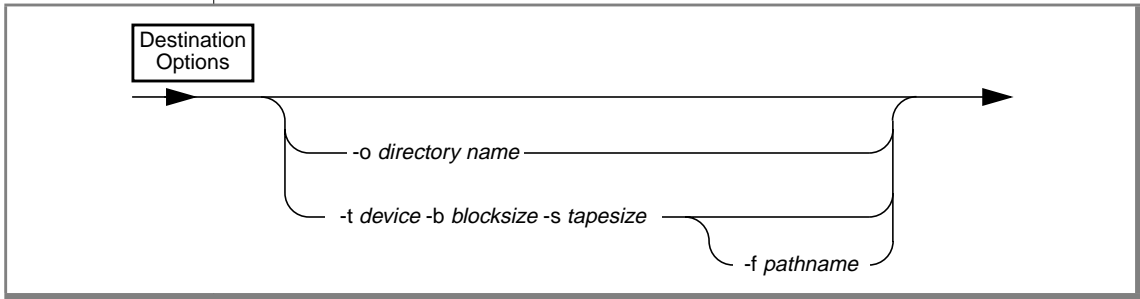
Using the -ss Option

The -ss option generates server-specific information. The -ss option specifies initial- and next-extent sizes, fragmentation information if the table is fragmented, the locking mode, the dbspace for a table, the blob space for any simple-large-objects, and the sb space for any smart-large-objects.

For INFORMIX-SE databases, the -ss option generates the pathname of each table that is in a path other than the database directory. ♦

SE

Destination Options



Element	Purpose	Key Considerations
<code>-b blocksize</code>	Specifies, in kilobytes, the block size of the tape device.	None.
<code>-f pathname</code>	Specifies the pathname where you want the schema file stored, if you are storing the data files on tape.	Additional Information: The pathname can be a complete pathname or simply a filename. If only a filename is given, the file is stored in the current directory.
<code>-o directory name</code>	Names the directory on disk where you want both the data files and the schema file stored.	Restrictions: The directory specified as <i>directory name</i> must already exist.
<code>-s tapesize</code>	Specifies, in kilobytes, the amount of data that you can store on the tape.	Restrictions: The tape size is limited to 2,097,151 kilobytes. The limit is required because of the way dbexport and dbimport track their positions into the tape.
<code>-t device</code>	Specifies the pathname of the tape device where you want the text files and, possibly, the schema file stored.	Restrictions: The <code>-t</code> option does not allow you to specify a remote tape device.

When you write to disk, **dbexport** creates a subdirectory, *database.exp*, in the directory that the `-o` option specifies. The **dbexport** utility creates a file, *tablename.unl*, for each table in the database. The schema file is written to the file *database.sql*. The *.unl* and *.sql* files are stored in the *database.exp* directory.

If you do not specify a destination for the data and schema files, the subdirectory *database.exp* is placed in the current working directory.

SE

When you write the data files to tape, you can use the **-f** option to store the schema file to disk. You are not required to name the schema file **database.sql**. You can give it any name.

The following **dbexport** command creates a **reports.exp** subdirectory in the current directory. It then unloads the **reports** database in the **turku** directory on the SE database server called **finland** and places the resulting files in the **reports.exp** directory.

```
dbexport //finland/turku/reports
```



UNIX

For an OnLine Dynamic Server, OnLine Workgroup Server, or Universal Server, the same command would be as follows:

```
dbexport //finland/reports
```

The following command exports the database **stores7** to tape with a block size of 16 kilobytes and a tape capacity of 24,000 kilobytes. The schema file is written to **/tmp/stores7.imp**.

```
dbexport -t /dev/rmt0 -b 16 -s 24000 -f /tmp/stores7.imp  
stores7
```

The following command exports the same **stores7** database to the directory named **/work/exports/stores7.exp**. The resulting schema file is **/work/exports/stores7.exp/stores7.sql**.

```
dbexport -o /work/exports stores7
```



NT

For Windows NT, the following command exports the database **stores7** to tape with a block size of 16 kilobytes and a tape capacity of 24,000 kilobytes. The schema file is written to **C:\temp\stores7.imp**.

```
dbexport -t \\.\TAPEDRIVE -b 16 -s 24000 -f  
C:\temp\stores7.imp stores7
```

The following command exports the same **stores7** database to the directory named **D:\work\exports\stores7.exp**. The resulting schema file is **D:\work\exports\stores7.exp\stores7.sql**.

```
dbexport -o D:\work\exports stores7
```



IUS

The Contents of the Schema File

The schema file contains the SQL statements that you need to re-create the exported database. You can edit the schema file to modify the schema of the database.

The schema file supports all Universal Server data types. ♦

If you use the **-ss** option, the schema file contains server-specific information, such as initial- and next-extent sizes, fragmentation information, lock mode, the dbspace where each table resides, the blobspace where each simple-large-object column resides, and the sbspace for smart-large-objects. The following information is not retained:

- Logging mode of the database (For information about logging modes, refer to the [Informix Guide to SQL: Reference](#).)
- The starting values of SERIAL columns

The statements in the schema file that create tables, views, indexes, roles, and grant privileges do so with the name of the user who originally created the database. In this way, the original owner retains DBA privileges for the database and is the owner of all the tables, indexes, and views. In addition, the person who executes the **dbimport** command also has DBA privileges for the database.

Important: Do not put comments into the schema file. Comments cause unpredictable results when the **dbimport** utility uses the schema file.



NT

Exporting Simple Large Objects

When **dbimport**, **dbexport**, and DB-Access process blob data (simple large objects), they create temporary files for that data. Before you export or import data from tables that contain simple large objects, you must have one of the following:

- A **\tmp** directory on your currently active drive
- The **DBTEMP** environment variable set to point to a directory that is available for temporary storage of the simple large objects

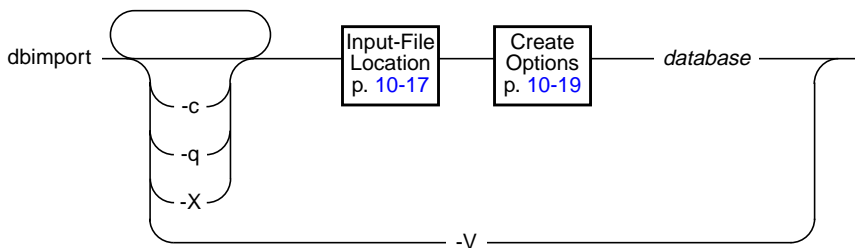
Windows NT sets the TMP and TEMP environment variables in the command prompt sessions, by default. However, if the TMP, TEMP, and DBTEMP environment variables are not set, **dbimport** places the temporary files for the simple large objects in the \tmp directory. ♦



The dbimport Utility

The **dbimport** utility creates a database and loads it with data from text files. The input files consist of a schema file that is used to re-create the database and data files that contain the database data. Normally, you generate the input files with the **dbexport** utility, but you can use any properly formatted input files. The **dbimport** utility supports Universal Server data types.

Tip: The examples in this chapter apply to Universal Server, OnLine Dynamic Server, and OnLine Workgroup Server unless marked with an icon for SE.



Element	Purpose	Key Considerations
-c	Instructs dbimport to complete importing even though it encounters certain nonfatal errors.	References: For more information, refer to “Using the -c Option” on page 10-16.
-q	Suppresses the display of error messages, warnings, and generated SQL data-definition statements.	None.
-V	Displays product version information.	None.
-X	Recognizes HEX binary data in character fields.	None.
database	Specifies the name of the database to create.	Additional Information: If you want to use more than the simple name of the database, refer to the Database Name section of the Informix Guide to SQL: Syntax.



SE

GLS

The **dbimport** utility can use files from the following location options:

- All input files are located on disk.
- All input files are located on tape.
- The schema file is located on disk, and the data files are located on tape.

***Important:** Do not put comments into your input file. Comments cause unpredictable results when the **dbimport** utility reads them.*

The **dbimport** utility supports the following options for a new Universal Server, OnLine Dynamic Server, or OnLine Workgroup Server database:

- Create an ANSI-compliant database (includes unbuffered logging).
- Establish transaction logging for a database (unbuffered or buffered logging).
- Specify the dbspace where the database will reside.

The **dbimport** utility supports the following options for a new SE database:

- Create an ANSI-compliant database (ANSI-compliant logging).
- Establish transaction logging for a database (unbuffered logging). ♦

The user who runs **dbimport** is granted the DBA privilege on the newly created database. The **dbimport** process locks each table as it is being loaded and unlocks the table when the loading is completed.

When the GLS environment variables are set correctly, as described in the [Guide to GLS Functionality](#), **dbimport** can import the data into Universal Server and Version 7.2x databases. ♦

Using the -c Option

If you include the **-c** option, **dbimport** ignores the following errors:

- A data row that contains too many columns
- Inability to put a lock on a table
- Inability to release a lock

Even if you use the **-c** option, **dbimport** interrupts processing if one of the following fatal errors occurs:

- Unable to open the tape device specified
- Bad writes to the tape or disk
- Invalid command parameters
- Cannot open database or no system permission
- Cannot convert the data

The **dbimport** utility creates a file of messages called **dbimport.out** in the current directory. This file contains any error messages and warnings that are related to **dbimport** processing. The same material is also written to the standard output unless you specify the **-q** option.

Using Delimited Identifiers with dbimport

You can use delimited identifiers with the **dbimport** utility. The utility detects database objects that are keywords, mixed case, or have special characters and places double quotes around them.

Canceling dbimport

You can press the INTERRUPT key at any time to cancel **dbimport**. The **dbimport** program asks for confirmation before it terminates.

Database-Logging Mode

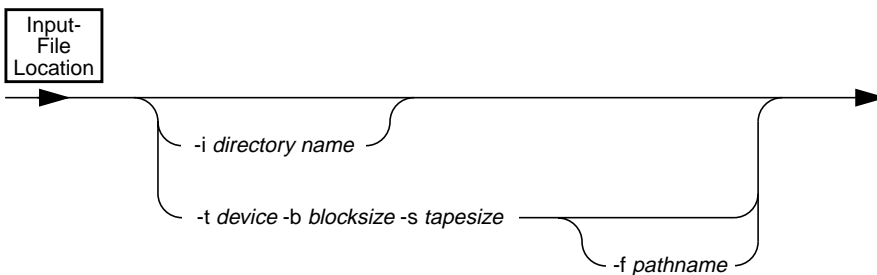
The logging mode is not retained in the schema file. You can specify any of the following options when you use **dbimport** to import a database:

- ANSI-compliant database with unbuffered logging
- Unbuffered logging
- Buffered logging

For more information, refer to [“Create Options” on page 10-19](#).

Input-File Location Options

The input-file location tells **dbimport** where to look for the files that it will import. If you do not specify an input-file location, **dbimport** looks for data files in the directory *database.exp* under the current directory and for the schema file in *database.exp/database.sql*.



Input-File Location Options

Element	Purpose	Key Considerations
-b <i>blocksize</i>	Specifies, in kilobytes, the block size of the tape device.	Restrictions: If you are importing from tape, you must use the same block size that you used to export the database.
-f <i>pathname</i>	Specifies where dbimport can find the schema file to use as input to create the database when the data files are read from tape.	Additional Information: If you use the -f option to export a database, you typically use the same pathname that you specified in the dbexport command. If you specify only a filename, dbimport looks for the file in the .exp subdirectory of your current directory.
-i <i>directory name</i>	Specifies the complete pathname of the directory on disk that contains the input data files and schema file. The directory name should be the same as the database name.	Additional Information: This directory should be the same directory that you specified with the dbexport -o option. If you change the directory name, you also rename your database.
-s <i>tapesize</i>	Specifies, in kilobytes, the amount of data that you can store on the tape.	Restrictions: If you are importing from tape, you must use the same tape size that you used to export the database.
-t <i>device</i>	Specifies the pathname of the tape device that holds the input files.	Restrictions: The -t option does <i>not</i> allow you to specify a remote tape device.

UNIX

The following command imports the **stores7** database from a tape with a block size of 16 kilobytes and a capacity of 24,000 kilobytes. The schema file is read from **/tmp/stores7.imp**.

```
dbimport -c -t /dev/rmt0 -b 16 -s 24000 -f
/tmp/stores7.imp stores7
```

The following command imports the **stores7** database from the **stores7.exp** directory under the **/work/exports** directory. The schema file is assumed to be **/work/exports/stores7.exp/stores7.sql**.

```
dbimport -c -i /work/exports stores7
```

◆

NT

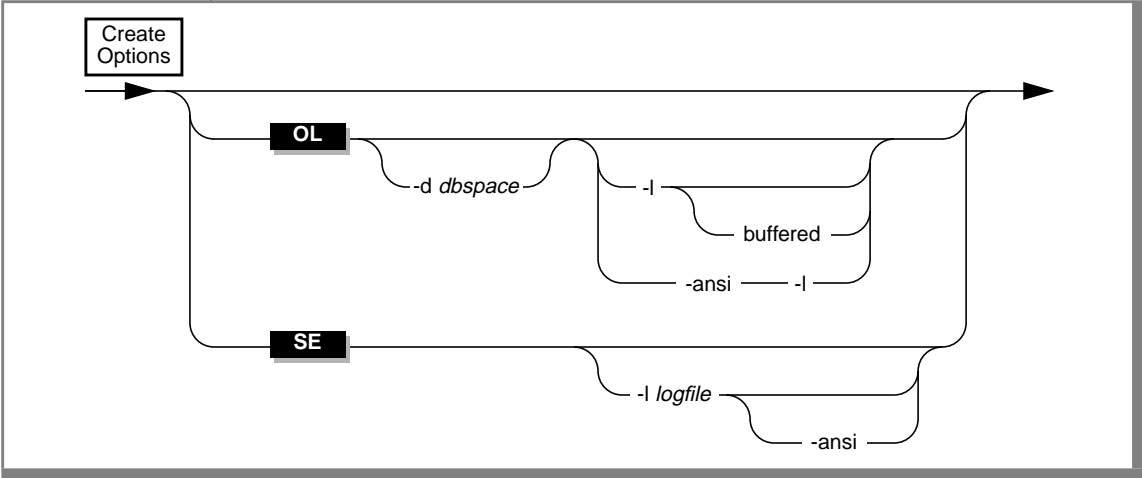
The following command imports the **stores7** database from a tape with a block size of 16 kilobytes and a capacity of 24,000 kilobytes. The schema file is read from **C:\temp\stores7.imp**.

```
dbimport -c -t \\.\TAPEDRIVE -b 16 -s 24000 -f
C:\temp\stores7.imp stores7
```

The following command imports the **stores7** database from the **stores7.exp** directory under the **D:\work\exports** directory. The schema file is assumed to be **D:\work\exports\stores7.exp\stores7.sql**.

```
dbimport -c -i D:\work\exports stores7
```

Create Options



Element	Purpose	Key Considerations
-ansi	Creates an ANSI-compliant database in which the ANSI rules for transaction logging are enabled.	Additional Information: If you specify the -ansi option, you must also specify the -l logfile option. For more information about ANSI-compliant databases, refer to the Informix Guide to SQL: Reference .
-d dbspace	Names the dbspace where the database is created. The default dbspace location is the rootdbs.	Additional Information: For SE, the database is always in the current directory.

Element	Purpose	Key Considerations
-l	Establishes unbuffered transaction logging for the imported database.	References: For more information, refer to “Using the -l Options” on this page.
-l buffered	Establishes buffered transaction logging for the imported database.	References: For more information, refer to “Using the -l Options.”.
-l logfile	Establishes transaction logging for the imported database and specifies the name of the transaction-log file.	Restrictions: For SE, the <i>logfile</i> filename must be an absolute pathname or in the current directory. References: For more information, refer to “Using the -l Options”.

(2 of 2)

UNIX

The following command imports the **stores7** database from the **/usr/informix/port/stores7.exp** directory to the current directory. The new database is ANSI compliant, and the transaction-log file is specified as **stores7.log** in **/usr/work**.

```
dbimport -c stores7 -i /usr/informix/port -l
/usr/work/stores7.log -ansi
```



NT

The following command imports the **stores7** database from the **C:\USER\informix\port\stores7.exp** directory to the current directory. The new database is ANSI compliant, and the transaction-log file is specified as **stores7.log** in **C:\USER\work**.

```
dbimport -c stores7 -i C:\USER\informix\port -l
C:\USER\work\stores7.log -ansi
```



Using the -l Options

The **-l** options are equivalent to the logging clauses of the CREATE DATABASE statement, as follows:

- The **-l** option is equivalent to the WITH LOG clause.
- The **-l buffered** option is equivalent to the WITH BUFFERED LOG.
- The **-l logfile** option is equivalent to the WITH LOG IN clause. ◆

For more information about the CREATE DATABASE statement, see Chapter 1 in the [Informix Guide to SQL: Syntax](#).

SE

Changing the Database Name

The **dbimport** utility assumes that the new database has the same name as the database that you exported. If you export a database to tape, you cannot change its name when you import it with **dbimport**.

If you export a database to disk, you can change the database name.

UNIX

To change the database name to **newname** on UNIX

In the following example, assume that **dbexport** unloaded the database **stores7** into the directory **/work/exports/stores7.exp**. Thus, the data files (the **.unl** files) are stored in **/work/exports/stores7.exp**, and the schema file is **/work/exports/stores7.exp/stores7.sql**.

1. Change the name of the **.exp** directory. That is, change **/work/exports/stores7.exp** to **/work/exports/newname.exp**.
2. Change the name of the schema file. That is, change **/work/exports/stores7.exp/stores7.sql** to **/work/exports/stores7.exp/newname.sql**. Do not change the names of the **.unl** files.
3. Import the database with the following command:

```
dbimport -i /work/exports
```



NT

To change the database name to **newname** on Windows NT

In the following example, assume that **dbexport** unloaded the database **stores7** into the directory **D:\work\exports\stores7.exp**. Thus, the data files (the **.unl** files) are stored in **D:\work\exports\stores7.exp**, and the schema file is **D:\work\exports\stores7.exp\stores7.sql**.

1. Change the name of the **.exp** directory. That is, change **D:\work\exports\stores7.exp** to **D:\work\exports\newname.exp**.
2. Change the name of the schema file. That is, change **D:\work\exports\stores7.exp\stores7.sql** to **D:\work\exports\stores7.exp\newname.sql**. Do not change the names of the **.unl** files.
3. Import the database with the following command:

```
dbimport -i D:\work\exports
```



NT

Importing Simple Large Objects

When **dbimport**, **dbexport**, and DB-Access process simple large objects (blobs), they create temporary files for that data. Before you export or import data from tables that contain simple large objects, you must have one of the following:

- A **\tmp** directory on your currently active drive
- The **DBTEMP** environment variable set to point to a directory that is available for temporary storage of the simple large objects

Windows NT sets the **TMP** and **TEMP** environment variables in the command prompt sessions, by default. However, if the **TMP**, **TEMP**, and **DBTEMP** environment variables are not set, **dbimport** places the temporary files for simple large objects in the **\tmp** directory. ♦

Using dbimport to Change the Locale of a Database

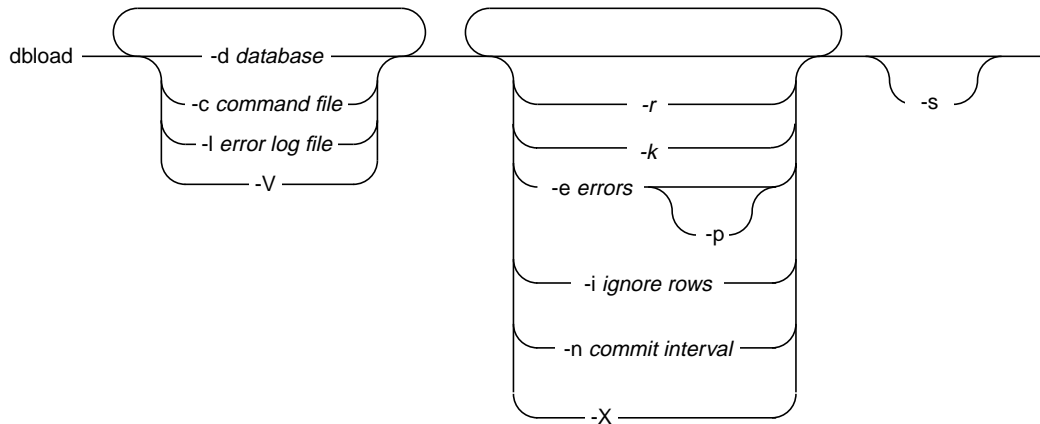
You can use **dbimport** to change the locale of a database.

To change the locale of a database

1. Run **dbexport** on the database.
2. Set the **DBLOCALE** environment variable to the desired locale for the new database.
3. Run **dbimport** to import the data.

The dbload Utility

The **dbload** utility transfers data from one or more text files into one or more existing tables. The **dbload** utility supports Universal Server data types.



Element	Purpose	Key Considerations
-c command file	Specifies the filename or pathname of a dbload command file.	References: For information about building the command file, refer to “Creating a dbload Command File” on page 10-27 .
-d database	Specifies the name of the database to receive the data.	Additional Information: If you want to use more than the simple name of the database, refer to the Database Name section of the Informix Guide to SQL: Syntax .
-e errors	Specifies the number of bad rows that dbload reads before terminating. The default value for <i>errors</i> is 10.	References: For more information, refer to “Using the -e and -p Options” on page 10-26 .
-i ignore rows	Specifies the number of rows to ignore in the input file.	References: For more information, refer to “Using the -i Option” on page 10-26 .

(1 of 2)

Element	Purpose	Key Considerations
-k	Instructs dbload to lock the tables listed in the command file in exclusive mode during the load operation.	<p>References: For more information, refer to “Using the -k Option” on page 10-25.</p> <p>Restrictions: You cannot use the -k option with the -r option because the -r option specifies that no tables are locked during the load operation.</p>
-l error log file	Specifies the filename or pathname of an error log file.	<p>Restrictions: If you specify an existing file, its contents are overwritten. If you specify a file that does not exist, dbload creates the file.</p> <p>Additional Information: The error log file stores diagnostic information and any input file rows that dbload cannot insert into the database.</p>
-n commit interval	Specifies the commit interval in number of rows. The default interval is 100 rows.	<p>Additional Information: If your database supports transactions, dbload commits a transaction after the specified number of new rows is read and inserted. A message appears after each commit.</p> <p>References: For information about transactions, see the Informix Guide to SQL: Tutorial.</p>
-p	Prompts for instructions if the number of bad rows exceeds the limit.	<p>References: For more information, refer to “Using the -e and -p Options” on page 10-26.</p>
-r	Prevents dbload from locking the tables during a load, thus enabling other users to update data in the table during the load.	<p>Additional Information: For more information, refer to “Using the -r Option” on page 10-26.</p> <p>Restrictions: You cannot use the -r option with the -k option because the -r option specifies that the tables are not locked during the load operation while the -k option specifies that the tables are locked in exclusive mode.</p>
-s	Checks the syntax of the statements in the command file without inserting data.	<p>Additional Information: The standard output displays the command file with any errors marked where they are found.</p>
-V	Displays product version information.	
-X	Recognizes HEX binary data in character fields.	

(2 of 2)



Tip: If you specify part (but not all) of the required information, **dbload** prompts you for additional specifications. If you are missing all three options, you receive an error message.

Using the dbload Utility

If you are on a network, you can specify a database on another database server by including the database server name and directory path with the database name.

You can load blobs with the **dbload** utility as long as the simple large objects are in text files.

The presence of indexes greatly affects the speed with which the **dbload** utility loads data. For best performance, drop any indexes on the tables that receive the data before you run **dbload**. You can create new indexes after **dbload** has finished.

You can use delimited identifiers with the **dbload** utility. The utility detects database objects that are keywords, mixed case, or have special characters, and places double quotes around them.

If your most-recent **dbload** session ended prematurely, you can resume loading with the next record in the file by specifying the starting line number in the command-line syntax.

If you press the INTERRUPT key, **dbload** terminates and discards any new rows that were inserted but not yet committed to the database (if the database has transactions).

Using dbload with SE

SE

The following command loads data into the **stores7** database in the **turku** directory on the SE database server **finland**:

```
dbload -d //finland/turku/stores7 -c commands -l errlog
```

◆

Using the -k Option

If you do not specify the **-k** option, the tables specified in the command file are locked in shared mode. When tables are locked in shared mode, the database server still has to acquire exclusive row or page locks when it inserts rows into the table.

When you specify the **-k** option, the database server places an exclusive lock on the entire table. The **-k** option increases performance for large loads because the database server does not have to acquire exclusive locks on rows or pages as it inserts rows during the load operation.

Using the -r Option

If you do not specify the **-r** option, the tables specified in the command file are locked during loading so that other users cannot update data in the table. Table locking reduces the number of locks needed during the load but reduces concurrency. If you are planning to load a large number of rows, use table locking and load during nonpeak hours.

However, you can override this default locking mode by specifying the **-k** option. The **-k** option instructs **dbload** to lock the tables in exclusive mode rather than shared mode during the load operation.

Using the -i Option

The **-i** option instructs **dbload** to read and ignore the specified number of new-line characters in the input file before it begins to process. This option is useful if your most-recent **dbload** session ended prematurely. For example, if **dbload** ends after it inserts 240 lines of input, you can begin to load again at line 241 if you set *number rows ignore* to 240. It is also useful if header information in the input file precedes the data records.

Using the -e and -p Options

The **-e** option lets you specify how many bad rows to allow before **dbload** terminates.

If you set *errors* to a positive integer, **dbload** terminates when it reads (*errors* + 1) bad rows. If you set *errors* to zero, **dbload** terminates when it reads the first bad row.

If **dbload** exceeds the bad-row limit and the **-p** option is specified, **dbload** prompts you for instructions before it terminates. The prompt asks whether you want to roll back or to commit all rows that were inserted since the last transaction.

If **dbload** exceeds the bad-row limit and the **-p** option is not specified, **dbload** commits all rows that were inserted since the last transaction.

Choosing Between dbload and LOAD

The **dbload** utility offers the following advantages over the LOAD statement:

- You can use **dbload** to load data from input files that have been created with a variety of format arrangements. The **dbload** command file can accommodate data from entirely different database management systems.
- You can specify a starting point in the load by directing **dbload** to read but ignore *x* number of rows.
- You can specify a batch size so that after every *x* number of rows are inserted, the insert is committed.
- You can limit the number of bad rows read, beyond which **dbload** ends.

The cost of **dbload** flexibility is the time and effort spent creating the **dbload** command file, which is required for **dbload** operation. The input files are not specified as part of the **dbload** command line, and neither are the tables into which the data is inserted. This information is contained in the command file.

Creating a dbload Command File

Before you use **dbload**, you must create a command file that names the input data files and the tables that receive the data. The command file maps fields from one or more input files into columns of one or more tables within your database.

The command file contains only FILE and INSERT statements. Each FILE statement names an input data file. The FILE statement also defines the data fields from the input file that are inserted into the table. Each INSERT statement names a table to receive the data. The INSERT statement also defines how **dbload** places the data that is described in the FILE statement into the table columns.

Within the command file, the FILE statement can appear in the following forms:

- Delimiter form
- Character-position form

The FILE statement has a size limit of 4,096 bytes.

Use the delimiter form of the FILE statement when every field in the input data row uses the same delimiter and every row ends with a new-line character. This format is typical of data rows with variable-length fields. You can also use the delimiter form of the FILE statement with fixed-length fields as long as the data rows meet the delimiter and new-line requirements. The delimiter form of the FILE and INSERT statements is easier to use than the character-position form.

Use the character-position form of the FILE statement when you cannot rely on delimiters and you need to identify the input data fields by character position within the input row. For example, use this form to indicate that the first input data field begins at character position 1 and continues until character position 20. You can also use this form if you must translate a character string into a null value. For example, if your input data file uses a sequence of blanks to indicate a null value, you must use this form if you want to instruct **dbload** to substitute null at every occurrence of the blank-character string.

You can use both forms of the FILE statement in a single command file. However, for clarity, the two forms are described separately in the following sections.

FILE and INSERT Statements: Delimiter Form

The following example of a **dbload** command file illustrates a simple delimiter form of the FILE and INSERT statements. The example is based on the **stores7** database. An UNLOAD statement created the three input data files, **stock.unl**, **customer.unl**, and **manufact.unl**. To see the **.unl** input data files, refer to the directory **\$INFORMIXDIR/demo/prod_name** (UNIX) or **%INFORMIXDIR%\demo\prod_name** (Windows NT).

```
FILE stock.unl DELIMITER '|' 6;
INSERT INTO stock;
FILE customer.unl DELIMITER '|' 10;
INSERT INTO customer;
FILE manufact.unl DELIMITER '|' 3;
INSERT INTO manufact;
```

Syntax for the Delimiter Form

The following diagram shows the syntax of the delimiter FILE statement.



Element	Purpose	Key Considerations
c	Defines the field delimiter for the specific input file.	Restrictions: If the delimiter specified by c appears as a literal character anywhere in the input file, the character must be preceded with a backslash in the input file. For example, if the value of c is specified as a square bracket (()), you must place a backslash before any literal [that appears in the input file. Similarly, you must precede any backslash that appears in the input file with an additional backslash.
filename	Specifies the input file.	None.
nfields	Indicates the number of fields in each data row.	None.

The **dbload** utility assigns the sequential names **f01**, **f02**, **f03**, and so on to fields in the input file. You cannot see these names, but if you refer to these fields to specify a value list in an associated INSERT statement, you must use the **f01**, **f02**, **f03** format. For details, refer to [“How to Write a dbload Command File in Delimiter Form” on page 10-31](#).

Two consecutive delimiters define a null field. As a precaution, you can place a delimiter immediately before the new-line character that marks the end of each data row. If you omit this delimiter, an error results whenever the last field of a data row is empty. If you are certain that none of the input data rows ends with an empty field, you can omit this step.

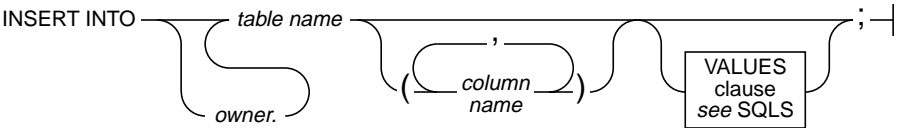
Inserted data types correspond to the explicit or default column list. If the data field width is different from its corresponding character column width, the data is made to fit. That is, inserted values are padded with blanks if the data is not wide enough for the column or truncated if the data is too wide for the column.

If the number of columns named is fewer than the number of columns in the table, **dbload** inserts the default value that was specified when the table was created for the unnamed columns. If no default value is specified, **dbload** attempts to insert a null value. If the attempt violates a not null restriction or a unique constraint, the insert fails, and an error message is returned.

If the INSERT statement omits the column names, the default INSERT specification is every column in the named table. If the INSERT statement omits the VALUES clause, the default INSERT specification is every field of the previous FILE statement.

An error results if the number of column names listed (or implied by default) does not match the number of values listed (or implied by default).

The syntax of **dbload** INSERT statements resembles INSERT statements in SQL, except that in **dbload**, INSERT statements cannot incorporate SELECT statements. The following diagram shows the syntax of the **dbload** INSERT statement for delimiter form.



Element	Purpose	Key Considerations
<i>column name</i>	Specifies the column that receives the new data.	None.
<i>owner.</i>	Specifies the user name of the table owner.	None.
<i>table name</i>	Specifies the table that receives the new data.	None.

Users who execute **dbload** with this command file must have the Insert privilege on the named table.

How to Write a dbload Command File in Delimiter Form

The first FILE and INSERT statement set in the delimiter example on [page 10-28](#) is repeated in the following example:

```
FILE stock.unl DELIMITER '|' 6;
INSERT INTO stock;
```

The FILE statement describes the **stock.unl** data rows as composed of six fields, each separated by a vertical bar (|) as the delimiter. Compare the FILE statement with the data rows in the following example, which appear in the input file **stock.unl**. (Because the last field is not followed by a delimiter, an error results if any data row ends with an empty field.)

```
1|SMT|baseball gloves|450.00|case|10 gloves/case
2|HRO|baseball|126.00|case|24/case
3|SHK|baseball bat|240.00|case|12/case
```

The example INSERT statement contains only the required elements. Because the column list is omitted, the INSERT statement implies that values are to be inserted into every field in the **stock** table. Because the VALUES clause is omitted, the INSERT statement implies that the input values for every field are defined in the most-recent FILE statement. This INSERT statement is valid because the **stock** table contains six fields, which is the same number of values that the FILE statement defines. The following example shows the first data row that is inserted into **stock** from this INSERT statement.

Field	Column	Value
f01	stock_num	1
f02	manu_code	SMT
f03	description	baseball gloves
f04	unit_price	450.00
f05	unit	case
f06	unit_descr	10 gloves/case



The FILE and INSERT statement in the following example illustrates a more complex INSERT statement syntax:

```
FILE stock.unl DELIMITER '|' 6;
INSERT INTO new_stock (col1, col2, col3, col5, col6)
VALUES (f01, f03, f02, f05, 'autographed');
```

In this example, the VALUES clause uses the field names that **dbload** assigns automatically. You must reference the automatically assigned field names with the letter **f** followed by a number: **f01, f02, f10, f100, f999, f1000**, and so on. All other formats are incorrect.

***Tip:** The first nine fields must include a zero: f01, f02, ..., f09.*

The user changed the column names, the order of the data, and the meaning of **col6** in the new **stock** table. Because the fourth column in **new_stock** (**col4**) is not named in the column list, the new data row contains a null in the **col4** position (assuming that the column permits nulls). If no default is specified for **col4**, the inserted value is null.

The following table shows the first data row that is inserted into **new_stock** from this INSERT statement.

Column	Value
col1	1
col2	baseball gloves
col3	SMT
col4	null
col5	case
col6	autographed

FILE and INSERT Statements: Character-Position Form

The examples in this section are based on an input data file, **cust_loc_data**, that contains the last four columns (**city**, **state**, **zipcode**, and **phone**) of the **customer** table. Fields in the input file are padded with blanks to create data rows in which the location of data fields and the number of characters are the same across all rows. The definitions for these fields are CHAR(15), CHAR(2), CHAR(5), and CHAR(12), respectively. Figure 10-1 displays the character positions and five example data rows from the **cust_loc_data** file.

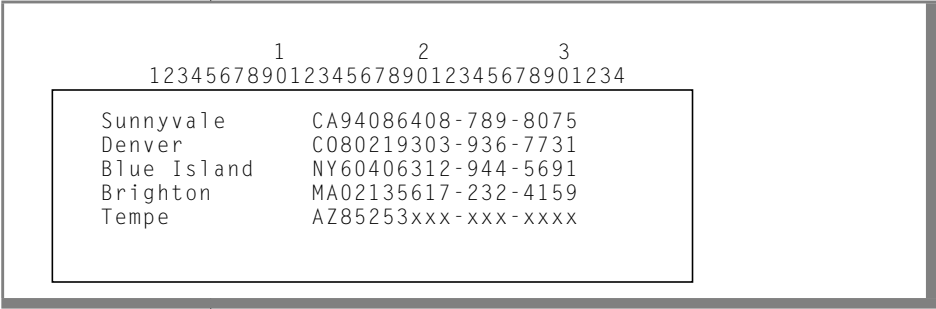


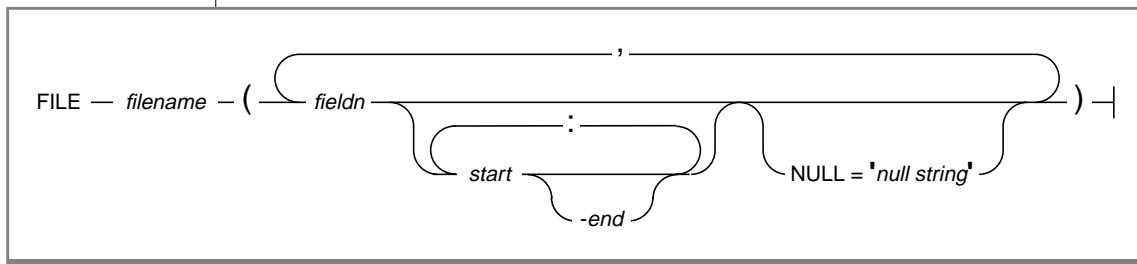
Figure 10-1
A Sample Data File

The following example of a **dbload** command file illustrates the character-position form of the **FILE** and **INSERT** statements. The example includes two new tables, **cust_address** and **cust_sort**, to receive the data. For the purpose of this example, **cust_address** contains four columns, the second of which is omitted from the column list. The **cust_sort** table contains two columns:

```
FILE cust_loc_data
  (city 1-15,
   state 16-17,
   area_cd 23-25 NULL = 'xxx',
   phone 23-34 NULL = 'xxx-xxx-xxxx',
   zip 18-22,
   state_area 16-17 : 23-25);
INSERT INTO cust_address (col1, col3, col4)
  VALUES (city, state, zip);
INSERT INTO cust_sort
  VALUES (area_cd, zip);
```

Syntax for the Character-Position Form

The following diagram shows the syntax of the character-position FILE statement.



Element	Purpose	Key Considerations
<i>-end</i>	Indicates the character position within a data row that ends a range of character positions.	Restrictions: A hyphen must precede the end value.
<i>fieldn</i>	Assigns a name to the data field that you are defining with the range of character positions.	None.
<i>filename</i>	Specifies the name of the input file.	None.
<i>null string</i>	Specifies the data value for which dbload should substitute a null.	Restrictions: Must be a quoted string.
<i>start</i>	Indicates the character position within a data row that starts a range of character positions. If you use <i>start</i> without <i>end</i> , it represents a single character.	None.

You can repeat the same character position in a data-field definition or in different fields.

The *null string* scope of reference is the data field for which you define it. You can define an explicit null string for each field that allows null entries.

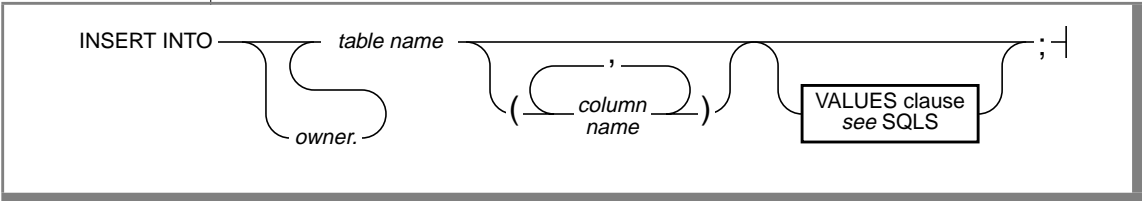
Inserted data types correspond to the explicit or default column list. If the data-field width is different from its corresponding character column, inserted values are padded with blanks if the column is wider or are truncated if the field is wider.

If the number of columns named is fewer than the number of columns in the table, **dbload** inserts the default value that is specified for the unnamed columns. If no default value is specified, **dbload** attempts to insert a null value. If the attempt violates a not null restriction or a unique constraint, the insert fails, and an error message is returned.

If the INSERT statement omits the column names, the default INSERT specification is every column in the named table. If the INSERT statement omits the VALUES clause, the default INSERT specification is every field of the previous FILE statement.

An error results if the number of column names listed (or implied by default) does not match the number of values listed (or implied by default).

The syntax of **dbload** INSERT statements resembles INSERT statements in SQL, except that in **dbload**, INSERT statements cannot incorporate SELECT statements. The following diagram shows the syntax of the **dbload** INSERT statement for character-position form.



Element	Purpose	Key Considerations
<i>column name</i>	Specifies the column that receives the new data.	None.
<i>table name</i>	Specifies the table that receives the new data.	None.
<i>owner.</i>	Specifies the user name of the table owner.	None.

The syntax for character-position form is identical to the syntax for delimiter form.

The user who executes **dbload** with this command file must have the Insert privilege on the named table.

SE

In INFORMIX-SE, the **dbload** utility recognizes valid SE table references, including owner designations. That is, the owner name can precede the table name but the database server name or the database name cannot precede the table name. Valid table-name syntax is defined in detail in the [Informix Guide to SQL: Syntax](#). ♦

How to Write a dbload Command File in Character-Position Form

The first FILE and INSERT statement set in the character-position example on [page 10-33](#) is repeated in the following example:

```
FILE cust_loc_data
  (city 1-15,
   state 16-17,
   area_cd 23-25 NULL = 'xxx',
   phone 23-34 NULL = 'xxx-xxx-xxxx',
   zip 18-22,
   state_area 16-17 : 23-25);
INSERT INTO cust_address (col1, col3, col4)
VALUES (city, state, zip);
```

The FILE statement defines six data fields from the **cust_loc_data** table data rows. The statement names the fields and uses character positions to define the length of each field. Compare the FILE statement in the preceding example with the data rows in Figure 10-2.

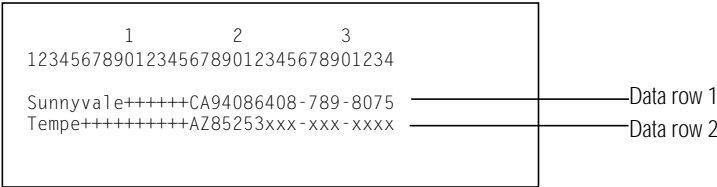


Figure 10-2
A Sample Data File

The FILE statement defines the following data fields, which are derived from the data rows in [Figure 10-2](#).

Column	Values from Data Row 1	Values from Data Row 2
city	Sunnyvale++++++	Tempe++++++++++
state	CA	AZ
area_cd	408	null
phone	408-789-8075	null
zip	94086	85253
state_area	CA408	AZxxx

The null strings that are defined for the **phone** and **area_cd** fields generate the null values in those columns but do not affect the values that are stored in the **state_area** column.

The INSERT statement uses the field names and values that are derived from the FILE statement as the value-list input. Consider the following INSERT statement:

```
INSERT INTO cust_address (col1, col3, col4)
VALUES (city, state, zip);
```

The INSERT statement uses the data shown in Figure 10-2 and the FILE statement on [page 10-36](#) to put the following information into the **cust_address** table.

Column	Values from Data Row 1	Values from Data Row 2
col1	Sunnyvale++++++	Tempe++++++++++
col2	null	null
col3	CA	AZ
col4	94086	85253

Because the second column (**col2**) in **cust_address** is not named, the new data row contains a null (assuming that the column permits nulls).

Consider the following INSERT statement:

```
INSERT INTO cust_sort
VALUES (area_cd, zip);
```

This INSERT statement inserts the following data rows into the **cust_sort** table.

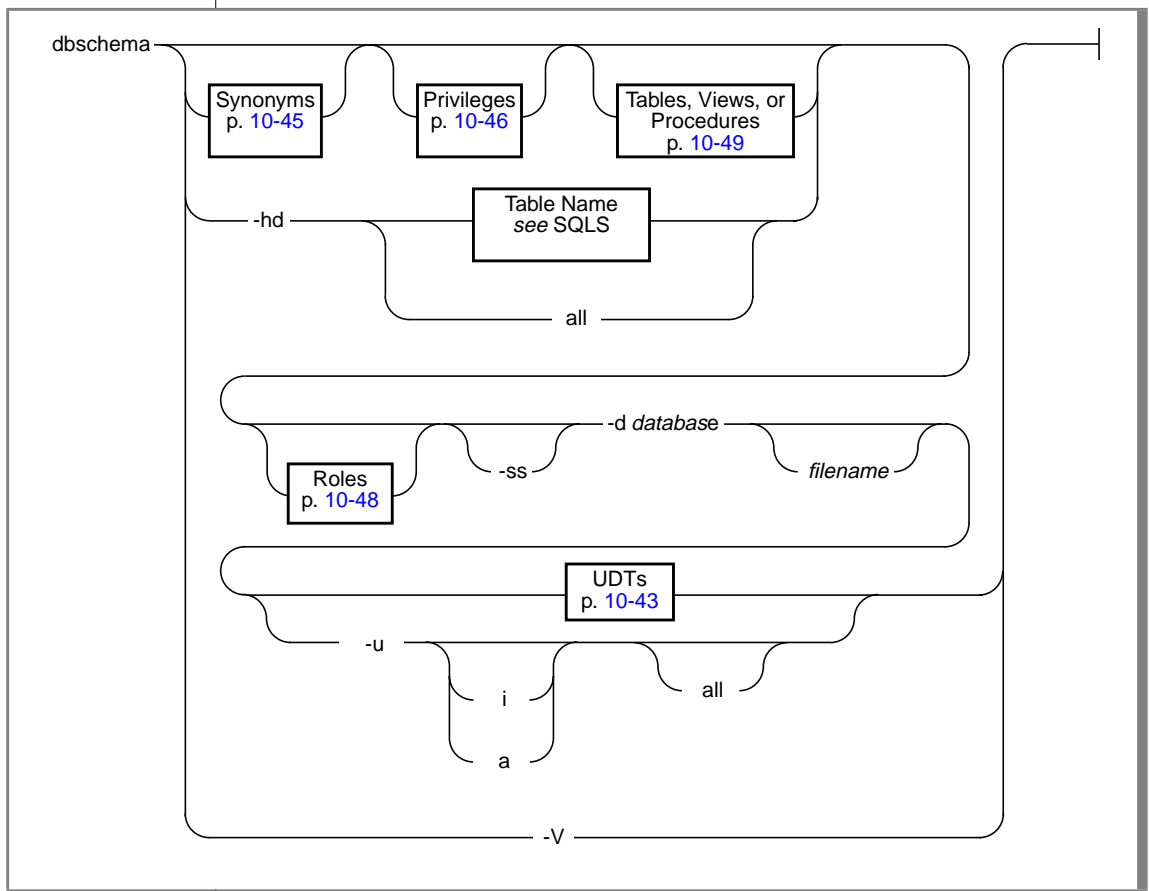
Column	Values from Data Row 1	Values from Data Row 2
col1	408	null
col2	94086	85253

Because no column list is provided, **dbload** reads the names of all the columns in **cust_sort** from the system catalog. (You cannot insert data into a temporary table because temporary tables are not entered into the system catalog.) Field names from the previous FILE statement specify the values to load into each column. You do not need one FILE statement for each INSERT statement.

The dbschema Utility

You can use the **dbschema** utility for the following purposes:

- To display the SQL statements (the *schema*) that are required to replicate a database or a specific table, view, or procedure
- To display the schema for the Information Schema views
- To display the distribution information that is stored for one or more tables in the database
- To display information on user-defined and row data types.



Element	Purpose	Key Considerations
-d database	Specifies the database to which the schema applies. The database can be on a remote database server.	References: If you want to use more than the simple name of the database, refer to the Database Name section of the Informix Guide to SQL: Syntax .
-hd	Displays the distribution as data values.	References: For more information, refer to “Displaying the Distribution Information for Tables” on page 10-51 .
-ss	Generates server-specific information.	Restrictions: This option is ignored if no table schema is generated. References: For more information, refer to “Using the -ss Option” on page 10-43 .
-u	Prints user-defined data types.	References: For more information, refer to “Using the -u Option” on page 10-43 .
-ui	Prints user-defined data types, including type inheritance.	References: For more information, refer to “Using the -u Option” on page 10-43 .
-ua	Prints user-defined data types including all functions and casts defined over a type.	References: For more information, refer to “Using the -u Option” on page 10-43 .
-V	Displays product version information.	None.
all	Directs dbschema to include all the tables in the database in the display of distributions.	None.
<i>filename</i>	Specifies the filename to contain the dbschema output.	Additional Information: If you do not supply a <i>filename</i> , dbschema sends output to the screen. If you do supply a <i>filename</i> , dbschema creates a file to contain the dbschema output and gives it the name you specify.

You must be the DBA or have the Connect or Resource privilege to the database before you can run **dbschema** on it.

SE

If you are using INFORMIX-SE, the database must exist in your current directory or in a directory that is cited in your **DBPATH** environment variable. ♦

GLS

When the GLS environment variables are set correctly, as described in the [Guide to GLS Functionality](#), **dbschema** can handle foreign characters and Version 7.2x and Universal Server databases. ♦



You can use delimited identifiers with the **dbschema** utility. The utility detects database objects that are keywords, mixed case, or have special characters and places double quotes around them.

Important: A **dbschema** run on a database server is not compatible with earlier versions of that database server. If you revert to an earlier version, edit the schema file to remove SQL statements that the earlier version does not support. For details about supported SQL statements, refer to the “[Informix Guide to SQL: Syntax](#).”

Creating the Schema for a Database

You can create the schema for an entire database or for a portion of the database. The options for **dbschema** allow you to perform the following actions:

- Display CREATE SYNONYM statements by owner, for a specific table or for the entire database.
- Display the CREATE TABLE, CREATE VIEW, CREATE FUNCTION, or CREATE PROCEDURE statements for a specific table or for the entire database.
- Display all GRANT privilege statements that affect a specified user or that affect all users for a database or a specific table. The user can be either a user name or role name.
- Display user-defined and row data types with or without type inheritance and casts. ♦

IUS

When you use **dbschema** and specify only the database name, it is equivalent to using **dbschema** with all its options (except for the **-hd** and **-ss** options). In addition, if Information Schema views were created for the database, this schema is shown. For example, the following two commands are equivalent:

```
dbschema -d stores7
dbschema -s all -p all -t all -f all -d stores7
```

The SERIAL fields included in CREATE TABLE statements displayed by **dbschema** do not specify a starting value. New SERIAL fields created with the schema file have a starting value of 1, regardless of their starting value in the original database. If this value is not acceptable, you must modify the schema file.

UNIX

Creating Schemas for Databases Across a Network

You can specify a database on any accessible Universal Server, OnLine Dynamic Server, or OnLine Workgroup Server with the **-d database** syntax. The following command displays the schema for the **stores7** database on the **finland** database server on the UNIX system console:

```
dbschema -d //finland/stores7
```

◆

SE

You can specify a database on another SE database server by including the database server name and directory path with the database name. The command in the following example displays the schema for the **stores7** database in the **turku** directory on the **finland** database server on the system console:

```
dbschema -d //finland/turku/stores7
```

◆

Owner Naming with *dbschema*

The **dbschema** utility uses the *owner.object* convention when it generates any CREATE TABLE, CREATE INDEX, CREATE SYNONYM, CREATE VIEW, CREATE PROCEDURE, CREATE FUNCTION, or GRANT statements, and when it reproduces any unique, referential, or check constraints. As a result, if you use the **dbschema** output to create a new object (table, index, view, procedure, constraint, or synonym), the owner of the original object owns the new object. If you want to change the owner of the new object, you must edit the **dbschema** output before you run it as an SQL script.

You can use the output of **dbschema** to create a new function if you also specify the *pathname* to a file in which compile-time warnings are stored. This pathname is displayed in the **dbschema** output.

For more information about the CREATE TABLE, CREATE INDEX, CREATE SYNONYM, CREATE VIEW, CREATE PROCEDURE, CREATE FUNCTION, and GRANT statements, see the [Informix Guide to SQL: Syntax](#).

Using the -ss Option

The **-ss** option generates server-specific information. In all Informix database servers except SE, the **-ss** option always generates the lock mode, extent sizes, and the dbspace name if the dbspace name is different from the database dbspace. In addition, if tables are fragmented, the **-ss** option displays information about the fragmentation strategy.

When you specify the **dbschema -ss** option, the output also displays any GRANT FRAGMENT statements that are issued for a particular user or in the entire schema.



Important: You must use the **dbschema -ss** option to get full information about the database. If you do not use **-ss**, you get only information about simple tables and columns, and no information about the fragmentation and storage options.

For more information about fragment-level authority, see the GRANT FRAGMENT and REVOKE FRAGMENT statements in the [Informix Guide to SQL: Syntax](#).

SE

In INFORMIX-SE, the **-ss** option generates the pathname where the table was created if the table is not in the database directory. ♦

IUS

Using the -u Option

When you specify the **dbschema -u** option, the output displays any user-defined and complex data types contained in the database. Two suboptions, **i** and **a**, let you display the type inheritance and all the functions and casts defined over a type respectively.

The following command displays all the user-defined and complex data types for the **stork** database:

```
dbschema -d stork -u all
```

Output from **dbschema** that is executed with the specified option `-u all` might appear as shown in the following example:

```
create row type 'informix'.person_t
(
    name varchar(30, 10) not null,
    address varchar(20, 10),
    city varchar(20, 10),
    state char(2),
    zip integer,
    bdate date
);
create row type 'informix'.employee_t
(
    salary integer,
    manager varchar(30, 10)
) under person_t;
```

The following command displays the user-defined and complex data types, as well as their type inheritance for the **person_t** table in the **stork** database:

```
dbschema -d stork -ui person_t
```

Output from **dbschema** executed with the option `-ui person_t` might appear as shown in the following example:

```
create row type 'informix'.person_t
(
    name varchar(30, 10) not null,
    address varchar(20, 10),
    city varchar(20, 10),
    state char(2),
    zip integer,
    bdate date
);
create row type 'informix'.employee_t
(
    salary integer,
    manager varchar(30, 10)
) under person_t;
create row type 'informix'.sales_rep_t
(
    rep_num integer,
    region_num integer,
    commission decimal(16),
    home_office boolean
) under employee_t;
```

The following command displays the user-defined and complex data types including all functions and class defines over a type for the **per_udt** table in the **stork** database:

```
dbschema -d stork -ua per_udt
```

Output from **dbschema** executed with the option **-ua per_udt** might appear as shown in the following example:

```
create opaque type 'informix'.per_udt
(
    internallength=95,
    alignment=1
);

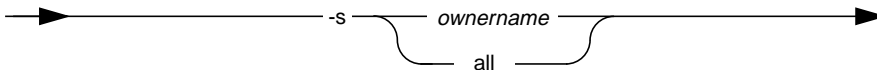
create implicit cast (lvarchar as per_udt with
'informix'.per_in);
CREATE FUNCTION "informix".per_in(l lvarchar)
    RETURNS per_udt
    EXTERNAL NAME '$USERFUNCDIR/person.so(Person_In)'
    LANGUAGE C ;

create implicit cast (lvarchar as per_udt with
'informix'.per_out);
CREATE FUNCTION "informix".per_out(p per_udt)
    RETURNS lvarchar
    EXTERNAL NAME '$USERFUNCDIR/person.so(PERSON_OUT)'
    LANGUAGE C NOT VARIANT;
```

◆

Obtaining the Synonym Schema

Synonyms



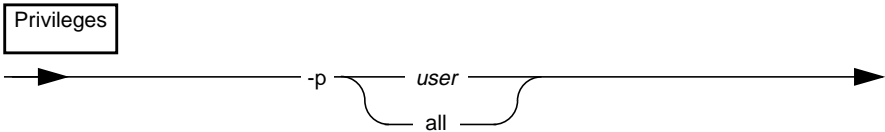
Element	Purpose	Key Considerations
-s <i>ownername</i>	Displays the CREATE SYNONYM statements owned by <i>ownername</i> .	None.
-s all	Displays all CREATE SYNONYM statements for the database, table, or view specified.	None.

Output from **dbschema** that is executed with the specified option -s *alice* might appear as shown in the following example:

```
CREATE SYNONYM 'alice'.cust FOR 'alice'.customer
```

For more information about the CREATE SYNONYM statement, see the [Informix Guide to SQL: Syntax](#).

Obtaining the Privilege Schema



Element	Purpose	Key Considerations
-p <i>user</i>	Displays the GRANT statements that grant privileges to a user where <i>user</i> can be a user name or role name. Specify only one user or role.	Restriction: You cannot specify a specific list of users with the -p option. You can specify either one user or role, or all users and roles.
-p all	Displays the GRANT statements for all users for the database, table, or view specified, or to all roles for the table specified.	None.

You cannot specify a specific list of users with the **-p** option. You can specify either one user or all users. The output also displays any GRANT FRAGMENT statements that are issued for a particular user or role or the entire schema.

In the **dbschema** output, the AS keyword indicates the grantor of a GRANT statement. The following example output indicates that **norma** issued the GRANT statement:

```
GRANT ALL ON 'tom'.customer TO 'claire' AS 'norma'
```

When the GRANT and AS keywords appear in the **dbschema** output, you might need to grant privileges before you run the **dbschema** output as an SQL script. Referring to the previous example output line, the following conditions must be true before you can run the statement as part of a script:

- **norma** must have the Connect privilege to the database.
- **norma** must have all privileges WITH GRANT OPTION for the table **tom.customer**.

For more information about the GRANT, GRANT FRAGMENT, and REVOKE FRAGMENT statements, see the [Informix Guide to SQL: Syntax](#).

Displaying Privilege Information for a Role

A *role* is a classification with privileges on database objects granted to the role. The DBA can assign the privileges of a related work task, such as an engineer, to a role and then grant that role to users, instead of granting the same set of privileges to every user. After a role is created, the DBA can use the GRANT statement to grant the role to users or to other roles.

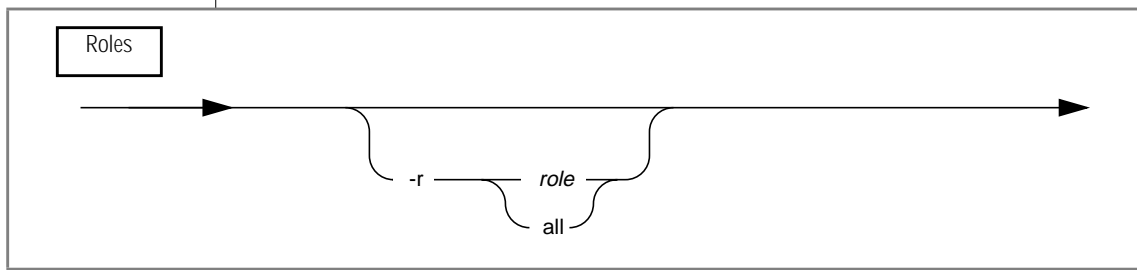
The following **dbschema** command and output show the privileges that were granted for the **calen** role:

```
sharky% dbschema -p calen -d stores7
```

```
DBSCHEMA Schema Utility      INFORMIX-SQL Version 7.22
Copyright (C) Informix Software, Inc., 1984-1996
Software Serial Number RDS#N000000
```

```
grant alter on table1 to 'calen'
```

Obtaining the Role Schema



Element	Purpose	Key Considerations
-r role	Displays the CREATE ROLE and GRANT statements that are needed to replicate and grant the specified role.	Restriction: You cannot specify a list of users or roles with the -r option. You can specify either one role, or all roles. SE does not support the -r option.
-r all	Displays all CREATE ROLE and GRANT statements that are needed to replicate and grant all roles.	None.

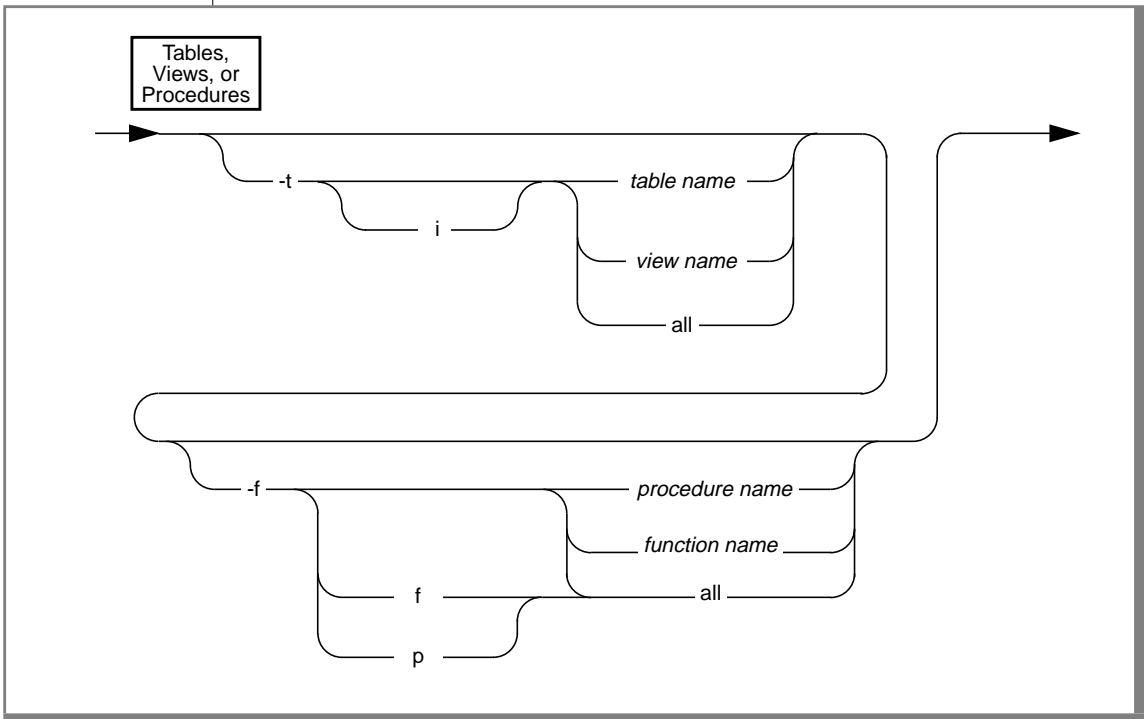
Displaying Information on Roles

The following **dbschema** command and output show that the role **calen** was created and was granted to **cathl**, **judith**, and **sallyc**:

```
sharky% dbschema -r calen -d stores7
```

```
DBSCHEMA Schema Utility          INFORMIX-SQL Version 7.22
Copyright (C) Informix Software, Inc., 1984-1996
Software Serial Number RDS#N000000
create role calen;

grant calen to cathl with grant option;
grant calen to judith ;
grant calen to sallyc ;
```


Specifying a Table, View, or Procedure

Element	Purpose	Key Considerations
-f all	Limits the SQL statement output to those statements that are needed to replicate all functions and procedures.	None.
-f function name	Limits the SQL statement output to only those statements that are needed to replicate the specified function.	None.
-f procedure name	Limits the SQL statement output to only those statements that are needed to replicate the specified procedure.	None.

(1 of 2)

Element	Purpose	Key Considerations
-ff all	Limits the SQL statement output to those statements that are needed to replicate all functions.	None.
-fp all	Limits the SQL statement output to those statements that are needed to replicate all procedures.	None.
-t table name	Limits the SQL statement output to only those statements that are needed to replicate the specified table.	None.
-t view name	Limits the SQL statement output to only those statements that are needed to replicate the specified view.	None.
-t all	Includes in the SQL statement output all statements that are needed to replicate all tables and views.	None.
-ti table name	Includes in the SQL statement output all statements that are needed to replicate all table levels.	None.
-ti all	Includes in the SQL statement output all statements that are needed to replicate all tables and views. Functionally equivalent to -t all .	None.

(2 of 2)

For more information about the CREATE PROCEDURE and CREATE FUNCTION statements, see the [Informix Guide to SQL: Syntax](#).

Using the -ss Option to Obtain Table Information

When you use the **-ss** option, you can retrieve information about fragmented tables, the lock mode, and extent sizes.

The following **dbschema** output shows the expressions specified for fragmented table.

```
DBSCHEMA Schema Utility      INFORMIX-SQL Version 7.20.UC1
Copyright (C) Informix Software, Inc., 1984-1995
{ TABLE "sallyc".t1 row size = 8 number of columns = 1 index size = 0 }
create table "sallyc".t1
(
  c1 integer
) fragment by expression
(c1 < 100 ) in db1 ,
((c1 >= 100 ) AND (c1 < 200 ) ) in db2 ,
remainder in db4
extent size 16 next size 16 lock mode page;
revoke all on "sallyc".t1 from "public";
```

Displaying the Distribution Information for Tables

To display the distribution information that is stored for a table in a database, use the **-hd** option with the name of the table. If you specify the **ALL** keyword for the table name, the distributions for all the tables in the database are displayed.

Distribution information is stored only if you have run the **UPDATE STATISTICS...MEDIUM** or **HIGH** statement for one or more columns of a table. For information about the **UPDATE STATISTICS** statement, refer to the [Informix Guide to SQL: Syntax](#).

The output of **dbschema** for distributions is provided in the following parts:

- Distribution description
- Distribution information
- Overflow information

Each section of **dbschema** output is explained in the following sections. As an example, the discussion uses the following distribution for the fictional table called **invoices**. This table contains 165 rows, including duplicates.

You can generate the output for this discussion with a call to **dbschema** that is similar to the following example:

```
dbschema -hd invoices -d pubs_stores7
```

Example Output

```
DBSCHEMA Schema Utility      INFORMIX-SQL Version 7.20.UC1
Copyright (C) Informix Software, Inc., 1984-1995
{

Distribution for cathl.invoices.invoice_num

Constructed on 03/10/1995

High Mode, 10.000000 Resolution

--- DISTRIBUTION ---

  (
1: ( 16,      7,      11)
2: ( 16,      6,      17)
3: ( 16,      8,      25)
4: ( 16,      8,      38)
5: ( 16,      7,      52)
6: ( 16,      8,      73)
7: ( 16,     12,      95)
8: ( 16,     12,     139)
9: ( 16,     11,     182)
10: ( 10,      5,     200)

--- OVERFLOW ---

  1: (   5,      56)
  2: (   6,      63)
}
```

Distribution Description

The first part of the **dbschema** output describes which data distributions have been created for the specified table. The name of the table is stated in the following example:

```
Distribution for cathl.invoices.invoice_num
```

The output is for the **invoices** table, which is owned by the user `cath1`. This data distribution describes the column **invoice_num**. If a table has distributions that are built on more than one column, **dbschema** lists the distributions for each column separately.

The date on which the distributions are constructed is listed. In this example, the date is `03/10/1995`, which is the date when the `UPDATE STATISTICS` statement that generated the distributions was executed. You can use this date to tell how outdated your distributions are. Although the system records the date, it does not record the time.

The last line of the description portion of the output describes the mode (medium or high) in which the distributions were created, and the resolution. If you create the distributions with medium mode, the confidence of the sample is also listed. For example, if the `UPDATE STATISTICS` statement is executed with high mode with a resolution of 10, the last line appears as shown in the following example:

```
High Mode, 10.000000 Resolution
```

The Distribution Information

The distribution information describes the bins that are created for the distribution, the range of values in the table and in each bin, and the number of distinct values in each bin. Consider the following example:

```
(
1: ( 16,      7,      11)
2: ( 16,      6,      17)
3: ( 16,      8,      25)
4: ( 16,      8,      38)
5: ( 16,      7,      52)
6: ( 16,      8,      73)
7: ( 16,     12,      95)
8: ( 16,     12,     139)
9: ( 16,     11,     182)
10: ( 10,      5,     200)
```

The first value in the rightmost column is the smallest value in this column. In this example, it is 5.

The column on the left shows the bin number, in this case 1 through 10. The first number in the parentheses shows how many values are in the bin. For this table, 10 percent of the total number of rows (165), is rounded down to 16. The first number is the same for all the bins except for the last. The last row might have a smaller value, indicating that it does not have as many row values. In this example, all the bins contain 16 rows except the last one, which contains 10.

The middle column within the parentheses indicates how many distinct values are contained in this bin. Thus, if there are 11 distinct values for a 16-value bin, it implies that one or more of those values are duplicated at least once.

The right column within the parentheses is the highest value in the bin. The highest value in the last bin is also the highest value in the table. For this example, the highest value in the last bin is 200.

The Overflow Information

The last portion of the **dbschema** output shows values that have many duplicates. The number of duplicates of indicated values must be greater than a critical amount that is determined as approximately 25 percent of the resolution times the number of rows. If left in the general distribution data, the duplicates would skew the distribution, so they are moved from the distribution to a separate list, as shown in the following example:

```
--- OVERFLOW ---  
1: ( 5, 56)  
2: ( 6, 63)
```

For this example, the critical amount is $0.25 * 0.10 * 165$, or 4.125. Therefore, any value that is duplicated five or more times is listed in the overflow section. Two values in this distribution are duplicated five or more times in the table: The value 56 is duplicated five times, and the value 63 is duplicated six times.

Object Modes and Violation Detection

The **dbschema** output supports object modes and violation detection:

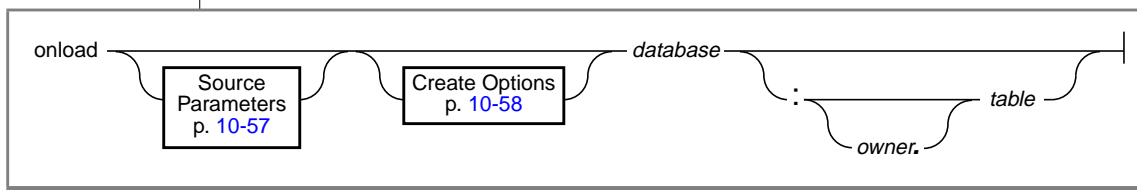
- The output shows the names of not-null constraints after the not-null specifications. You can use the output of the utility as input to create another database. If the same names were not used for not-null constraints in both databases, problems could result.
- The output shows the object mode of objects that are in the disabled state. These objects can be constraints, triggers, or indexes.
- The output shows the object mode of objects that are in the filtering state. These objects can be constraints or unique indexes.
- The output shows the violations and diagnostics tables that are associated with a base table (if violations and diagnostics tables have been started for the base table).

For more information about object modes and violation-detection, see the SET, START VIOLATIONS TABLE, and STOP VIOLATIONS TABLE statements in the [Informix Guide to SQL: Syntax](#).

The onload Utility

The **onload** utility creates a database or table in a specified Universal Server, OnLine Dynamic Server, or OnLine Workgroup Server dbspace. Then **onload** loads it with data from an input tape or disk file that the **onunload** utility creates.

During the load, you can move simple large objects that are stored in a blobspace to another blobspace.

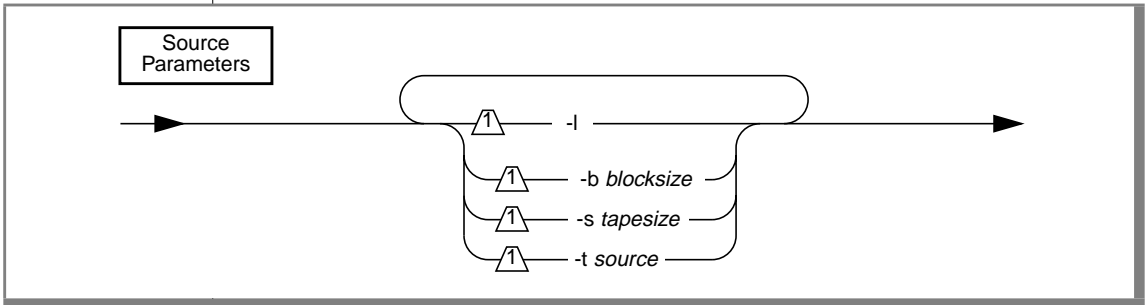


Element	Purpose	Key Considerations
<i>database</i>	Specifies the name of the database.	Restriction: The database name cannot include a database server name (<i>database@dbservername</i>). References: Syntax must conform to the Identifier segment; see the Informix Guide to SQL: Syntax .
<i>owner.</i>	Specifies the owner of the table.	Restriction: The owner name must not include illegal characters. References: For pathname syntax, refer to your operating-system documentation.
<i>table</i>	Specifies the name of the table.	Restriction: The table must not exist. References: Syntax must conform to the Table Name segment; see the Informix Guide to SQL: Syntax .

If you do not specify any source-parameter options, **onload** uses the device that is specified as TAPEDEV. The block size and tape size are the values that are specified as TAPEBLK and TAPESIZE, respectively. (For more information about TAPEDEV, TAPEBLK, and TAPESIZE, refer to the *INFORMIX-OnLine Dynamic Server Administrator's Guide*.)

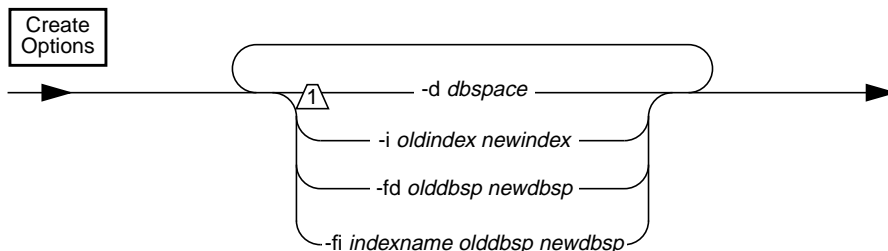
If you do not specify creation options, **onload** stores the database or table in the root dbspace.

Specify Source Parameters



Element	Purpose	Key Considerations
-b <i>blocksize</i>	Specifies in kilobytes the block size of the tape device.	Restrictions: Unsigned integer. Must specify the block size of the tape device. Additional Information: This option overrides the default value in TAPEBLK or LTAPEBLK.
-l	Directs onload to read the values for tape device, block size, and tape size from the configuration parameters LTAPEDEV, LTAPEBLK, and LTAPESIZE, respectively.	Additional Information: If you specify -l , and then -b , -s , or -t , the value that you specify overrides the value in the configuration file.
-s <i>tapesize</i>	Specifies in kilobytes the amount of data that the database server can store on the tape.	Restrictions: Unsigned integer. Must specify the amount of data that the database server can store on the tape. Additional Information: This option overrides the default value in TAPESIZE or LTAPESIZE.
-t <i>source</i>	Specifies the pathname of the file on disk or of the tape device where the input tape is mounted.	Restriction: Must be a legal pathname. Additional Information: This option overrides the tape device that TAPEDEV or LTAPEDEV specifies. References: For pathname syntax, see your operating-system documentation.

Create Options



Element	Purpose	Key Considerations
-d <i>dbspace</i>	Specifies the dbspace where the database or table will be stored.	Restriction: The dbspace must exist.
-fd <i>olddbspace newdbspace</i>	Allows you to move a data fragment from one dbspace to another.	Restriction: The new dbspace must exist and not already contain another data fragment for the table. Additional Information: This option is used with parallel data query (PDQ) and table fragmentation.
-fi <i>indexname olddbspace newdbspace</i>	Allows you to move index fragments from one dbspace to another.	Restriction: The new dbspace must exist and not already contain another index fragment for the table. Additional Information: This option is used with PDQ and table fragmentation.
-i <i>oldindex newindex</i>	Directs onload to rename the table index when it stores the index on disk.	Restriction: You must specify a table name in the command line. Additional Information: Use the -i option to rename indexes during the load to avoid conflict with existing index names. References: Syntax must conform to the Identifier segment; see the Informix Guide to SQL: Syntax .

If you do not specify any create options, the **onload** utility stores the database or table in the root dbspace.

You can use the **-d**, **-i**, **-fd**, and **-fi** options in any order and as often as necessary as long as you use unique pairs.

Constraints That Affect onload and onunload

The original database and the target database must be from the same version of Universal Server, OnLine Dynamic Server, or OnLine Workgroup Server. You cannot use the **onunload** and **onload** utilities to move data from one version to another. You also cannot use these utilities to move data between different types of servers.

You can use **onunload** and **onload** to move data between databases if the NLS and GLS locales are identical. For example, if user A has a French locale NLS table on server A, and then tries to load data into a German locale GLS table on server B, **onload/onunload** would report errors. However, if both the NLS and GLS tables were created with the same French locale, **onload/onunload** would work.

The tape that **onload** reads contains binary data that is stored in disk-page-sized units. For this reason, the computers where the original database resides (where you use **onunload**) and where the target database will reside (where you use **onload**) must have the following characteristics:

- The same page size
- The same representation of numeric data
- The same byte alignment for structures and unions

If the page sizes are different, **onload** fails. If the alignment or numeric data types on the two computers are different (for example, with the most-significant byte last instead of first, or different float-type representations), the contents of the data page could be misinterpreted.

Important: The **onunload** and **onload** utilities have the following restrictions:

- You cannot use **onunload** and **onload** to move data between non-GLS and GLS locales.
- Do not use **onunload** and **onload** to move data between two Universal Server databases if they contain extended data types. Use the HPL instead to move the Universal Server data. However, you can use **onunload** and **onload** with Universal Server if the databases contain legacy data types. ♦
- INFORMIX-SE does not support **onload** and **onunload**. ♦



IUS

SE

Constraints That Affect onload

The **onload** utility performs faster than the **dbimport**, **dbload**, or **LOAD** methods. In exchange for this higher performance, **onload** has the following constraints:

- The **onload** utility can only create a new database or table; you must drop or rename an existing database or table of the same name before you run **onload**. The **onload** utility prompts you to rename blobspaces during execution, if desired.
- The **onload** utility locks the database or table exclusively during the load.
- When you load a complete database, the user who executes **onload** becomes the owner of the database.
- The **onload** utility creates a database without logging; you must initiate logging after **onload** loads the database.
- When you use **onload** to load a table into a logged database, you must turn off logging for the database during the operation.

Logging While Using onload

The **onload** utility performs all of its loading within a transaction. This feature allows the changes to be rolled back if an error occurs.

When you use **onload** to create tables from an **onunload** input tape, **onload** can only load information into a database without logging. Thus, before you load a table into an existing, logged database, end logging for the database. You also might want to consider loading during off-peak hours. Otherwise, you might fill the logical-log files or consume excessive shared-memory resources. After you load the table, create a level-0 dbspace backup before you resume database logging.

When you use **onload** to create databases from an **onunload** input tape, the databases that result are not ANSI compliant and do not use transaction logging. You can make a database ANSI compliant, or add logging, after you load the database. (For more information about logging, refer to the [Informix Guide to SQL: Reference](#).)

Moving Simple Large Objects to a BlobSpace

If you load a table that contains simple large objects stored in a blobSpace, **onload** asks you if you want to move them to another blobSpace. If you respond *yes*, **onload** displays the blobSpace name where the simple large objects were stored when the tape was created. It then asks you to enter the name of the blobSpace where you want the simple large objects stored. If you enter a valid blobSpace name, **onload** moves all simple large object (TEXT and BYTE) columns in the table to the new blobSpace. Otherwise, **onload** prompts you again for a valid blobSpace name.

Ownership and Privileges

When you load a new database, the user who runs **onload** becomes the owner. Ownership within the database (tables, views, and indexes) remains the same as when the database was unloaded to tape with **onunload**.

To load a table, you must have Resource privilege on the database. When **onload** loads a new table, the user who runs **onload** becomes the owner unless you specify an owner in the table name. (To do so requires DBA privilege for the database.)

The **onunload** utility does not preserve synonyms or access privileges. To obtain a listing of defined synonyms or access privileges, use the **dbschema** utility, described on [page 10-39](#), before you run **onunload**.

Exclusive Locking During Load Operation

During the load operation, **onload** places an exclusive lock on the new database or table. Loading proceeds as a single transaction, and **onload** drops the new database or table if an error or system failure occurs.

The onmode Utility

This section discusses **onmode -b**. For information about the other **onmode** options, refer to the *Administrator's Guide* for your database server.

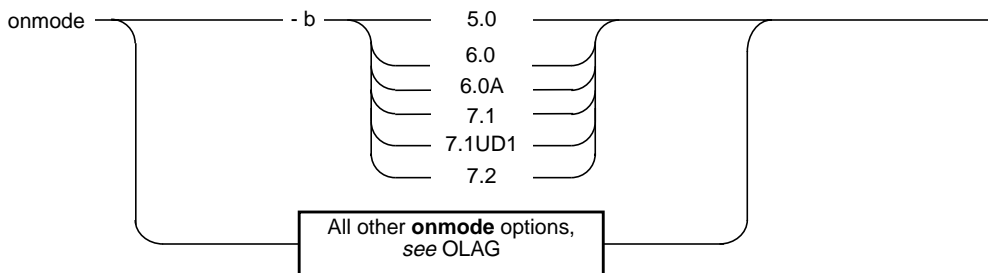
When you upgrade a database server, several modifications make the format of the database incompatible with the older version. The **onmode -b** command restores the databases to a format that is compatible with the earlier version. You must revert the databases before users can access the data with the earlier database server version. The utility does not revert changes made to the layout of the data that do not affect compatibility. ♦

You must be user **root** or user **informix** to execute **onmode**. ♦

You must be a member of the **Informix-Admin** group to execute **onmode**. ♦

Preparing to Use the -b Option

Before you use the **-b** option, notify users that you are going to bring the database server off-line. The reversion utility forcibly removes all users and shuts down the database server. The **-b** option includes an implicit **-yuk**. Make sure that the **INFORMIXSERVER** environment variable is set to the correct database server.



Element	Purpose	Key Considerations
-b 5.0	Change the database to the Version 5.0 format.	Additional Information: Refer to “Reverting to OnLine Version 5.0”.
-b 6.0	Change the database to the Version 6.0 format.	Additional Information: Refer to “Reverting to OnLine Dynamic Server Version 6.0” on page 10-65.
-b 6.0A	Change the database to the Version 6.0 ALS format.	Additional Information: Refer to “Reverting from Version 7.2x to Version 6.x ALS” on page 9-22.
-b 7.1	Change the database to the Version 7.1 format.	Additional Information: Refer to “Reverting to OnLine Dynamic Server Version 6.0, 7.1x, or OnLine Workgroup Server 7.12” on page 9-21 and “Reverting from OnLine Dynamic Server 7.1x to 7.1” on page 4-43.
-b 7.1UD1	Change the database to the Version 7.1UD1 format, which is compatible with 7.11, 7.12, 7.13, and 7.14 formats.	Additional Information: Refer to “Reverting to OnLine Dynamic Server Version 6.0, 7.1x, or OnLine Workgroup Server 7.12” on page 9-21.
-b 7.2	Change the database to the Version 7.2 format.	Additional Information: Refer to “Reverting to Version 7.2” on page 10-66 and “Reverting to OnLine Dynamic Server: UNIX” on page 4-33.

ODS

Reverting to OnLine Version 5.0

Before you use the **-b** option, you must free any resources that you allocated beyond Version 5.0 limits. Observe the following limits:

- The number of logical logs is less than or equal to 61.
- The number of dbspaces is less than or equal to 38.
- The number of chunks is less than or equal to 58.

In versions of OnLine Dynamic Server before Version 6.0, the maximum number of chunks depends on the length of the chunk pathnames but cannot exceed 58.

- Remove any fragmented tables.

Depending on the version of OnLine Dynamic Server from which you are reverting, the **onmode** utility performs the following actions:

- Verifies that no GLS databases exist
- Verifies that data replication is off

- Removes the second slot in the archive reserved page for data replication
- Drops the **sysmaster** database
- Rewrites leaf pages of all indexes
- Frees reserved-page extensions
- Removes the data-replication slot from the archive reserved page
- Rewrites all **partnums** on disk (systables, database tablespace, tablespace pages, blob freemap pages) in their old formats
- Rewrites dbspace page in the old format
- Writes a Version 5.0 format checkpoint record to a clean logical-log file

The reformatting does not make the data space identical to its earlier format. Some of the changes made during conversion from an earlier version to Version 7.2x do not make the space incompatible with earlier versions, and the **-b** option does not modify these changes. ♦

ODS

Reverting to OnLine Dynamic Server Version 6.0

Before you use the **-b** option, you must free any resources that you allocated beyond Version 6.0 limits. Observe the following limits:

- The number of page-cleaner threads is less than 33.
- The number of LRU queues is less than 32.

You set the maximum number of page-cleaner threads with the **CLEANER** configuration parameter and the maximum number of LRU queues with the **LRUS** configuration parameter.

Before you run **onmode -b 6.0**, you should also run **SINFORMIXDIR/etc/smi_unld** if either of the following conditions is true:

- You configured OnLine Dynamic Server for secure auditing.
- You used ON-Archive and want to preserve the associated catalog information.

Executing `$INFORMIXDIR/etc/smi_unld` preserves any data in the **sysmaster** database permanent tables. After you run **onmode -b**, OnLine Dynamic Server informs you of the **smi_load** script, which you can use to import your data for the permanent tables back into the **sysmaster** database once you initialize Version 6.0 of OnLine Dynamic Server.

The reversion from Version 7.2x to Version 6.0 performs the following actions:

- Verifies that none of the existing tables or indexes are fragmented
- Drops the **sysmaster** database ♦

Reverting to Version 7.2

Only Version 7.22 database servers support continuous data replication (Workgroup Replication or Enterprise Replication). Both Version 7.22 database servers and Universal Server (Version 9.1) support ON-Bar. Only Universal Server supports complex, user-defined, and built-in data types.

Execute **onmode -b 7.2** to revert the following:

- Version 7.22 to Version 7.2 or 7.21. The reversion utility automatically drops the **syscdr** database that continuous data replication uses. ♦
- Version 9.1 to Version 7.2x. The reversion utility automatically drops the new system catalog tables that Universal Server created as they are not used by OnLine Dynamic Server. After the system catalog tables are reverted, you can install either OnLine Dynamic Server or OnLine Workgroup Server 7.2, 7.21, or 7.22. ♦

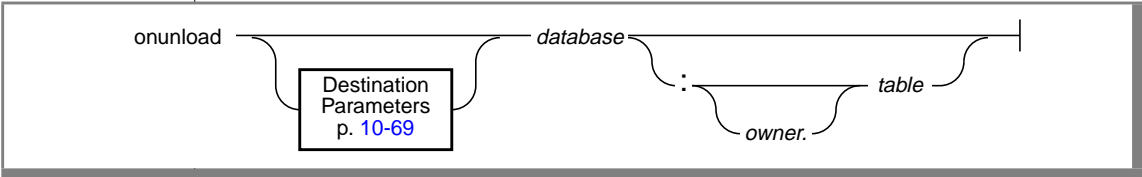
ODS, OWS

IUS

The onunload Utility

The **onunload** utility writes a database or table into a file on tape or disk. The **onunload** utility unloads the data in binary form in disk-page units, making this utility more efficient than **dbexport**. You frequently use the **onunload** utility to move data between computers.

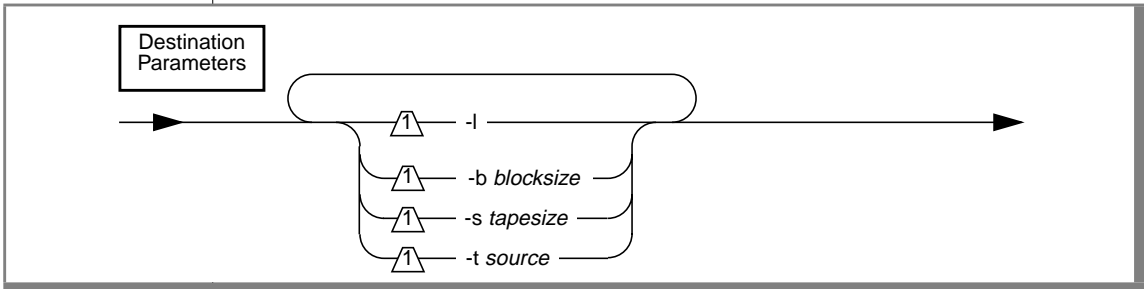
Syntax



Element	Purpose	Key Considerations
<i>database</i>	Specifies the name of a database.	Additional Information: The database name cannot include a database server name (<i>database@dbservername</i>). References: Syntax must conform to the Identifier segment; see the Informix Guide to SQL: Syntax .
<i>owner.</i>	Specifies the owner of the table.	Additional Information: The owner name must not include illegal characters. References: For pathname syntax, see your operating-system documentation.
<i>table</i>	Specifies the name of the table.	Restriction: The table must not exist. References: Syntax must conform to the Table Name segment; see the Informix Guide to SQL: Syntax .

If you do not specify any destination parameter options, **onunload** uses the device specified by TAPEDEV. The block size and tape size are the values specified as TAPEBLK and TAPESIZE, respectively. (For information about TAPEDEV, TAPEBLK, and TAPESIZE, refer to the Administrator’s Guide for your database server.)

Specify Destination Parameters



Element	Purpose	Key Considerations
-b <i>blocksize</i>	Specifies in kilobytes the block size of the tape device.	Restrictions: The <i>blocksize</i> must be an integer. Additional Information: This option overrides the default value in TAPEBLK or LTAPEBLK.
-l	Directs onunload to read the values for tape device, block size, and tape size from LTAPEDEV, LTAPEBLK, and LTAPESIZE, respectively.	None.
-s <i>tapesize</i>	Specifies in kilobytes the amount of data that can be stored on the tape.	Restrictions: The <i>tapesize</i> must be an integer. Additional Information: This option overrides the default value in TAPESIZE or LTAPESIZE.
-t <i>source</i>	Specifies the pathname of the file on disk or of the tape device where the input tape is mounted.	Additional Information: This option overrides the tape device specified by TAPEDEV or LTAPEDEV. It must be a legal pathname.

Constraints That Affect onunload

The **onunload** utility can unload data more quickly than either **dbexport** or the UNLOAD statement because it copies the data in binary and in page-sized units. However, this feature places the following constraints on its use:

- You must load the data on the **onunload** tape into a database or table managed by Universal Server, OnLine Dynamic Server, or OnLine Workgroup Server.
- You can use **onunload** and **onload** with Universal Server if the databases contain legacy data types.

- You must load the tape written by **onunload** onto a computer with the same page size and the same representation of numeric data as the original computer.
- You must read the file created by **onunload** with the **onload** utility of the same version of Universal Server, OnLine Dynamic Server, or OnLine Workgroup Server. You cannot use **onunload/onload** to move data from one version to another.
- When you unload a complete database, you cannot modify the ownership of database objects (such as tables, indexes, and views) until after you finish reloading the database.
- When you unload and load a table, **onunload** does not preserve access privilege, synonyms, views, constraints, triggers, or default values that were associated with the original tables. Before you run **onunload**, use the **dbschema** utility to obtain a listing of the access privilege, synonyms, views, constraints, triggers, or default values. After you finish loading the table, use **dbschema** to recreate the specific information for the table.

Unloading a Database or Table

To unload a database, you must have DBA privileges for the database or be user **informix**. To unload a table, you must either own the table, have DBA privileges for the database in which the table resides, or be user **informix**. (User **root** does not have special privileges with respect to **onunload** and **onload**.)

Unloading a Database

If you unload a database, all the tables in the database—including the system catalog tables—are unloaded. All the triggers, stored procedures, defaults, constraints, and synonyms for all the tables in the database are also unloaded.

Unloading a Table

If you unload a table, **onunload** unloads the table data and information from the following system catalog tables:

- **systables**
- **syscolumns**
- **sysindexes**
- **sysblobs**

When you unload a table, **onunload** does not unload information about constraints, triggers, or default values that are associated with a table. In addition, access privileges that are defined for the table and synonyms or views that are associated with the table are not unloaded.

Logging Mode

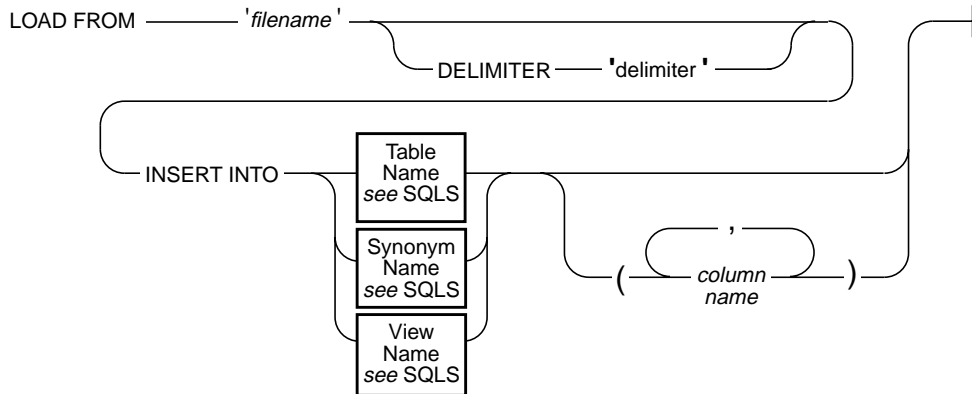
The **onunload** utility does not preserve the logging mode of a database. After you load the database with **onload**, you can make a database ANSI compliant or add logging. For information about logging modes, refer to the [Informix Guide to SQL: Syntax](#).

Locking During Unload Operation

During the unload operation, the database or table is locked in shared mode. An error is returned if **onunload** cannot obtain a shared lock.

The LOAD Statement

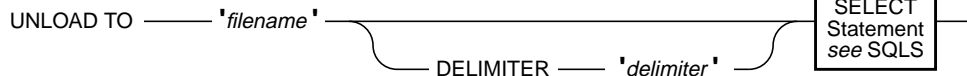
You can use the LOAD statement in DB-Access to append rows to an existing table of a database.



The preceding syntax diagram is included only for quick reference. For details about the syntax and use of the LOAD statement, refer to the [Informix Guide to SQL: Syntax](#).

The UNLOAD Statement

You can use the UNLOAD statement in DB-Access to unload selected rows from a database into a text file.



UNLOAD TO ——— 'filename' ——— DELIMITER ——— 'delimiter' ———

SELECT
Statement
see SQLS

The diagram shows the syntax of the UNLOAD statement. It starts with 'UNLOAD TO', followed by a line for the filename in single quotes. Then, there is a bracketed section for the delimiter, starting with 'DELIMITER', followed by a line for the delimiter in single quotes. To the right of this is a box containing the text 'SELECT Statement see SQLS', which is connected to the main line by a vertical line.

The preceding syntax diagram is included only for quick reference. For details about the syntax and use of the UNLOAD statement, refer to the [Informix Guide to SQL: Syntax](#).

ODS

UNIX

onpload: The High-Performance Loader Utility

The **onpload** utility is the command-line portion of the High-Performance Loader (HPL). You can use the HPL with OnLine Dynamic Server, Version 7.2 or later, and Universal Server. You cannot use the HPL with SE, OnLine Workgroup Server, or pre-7.2 versions of OnLine Dynamic Server.

For extremely large databases, the HPL has a performance advantage over other Informix data-migration utilities because it performs I/O and code set conversions in parallel. However, the user must invest significant preparation time before using the HPL and the HPL program itself has a significant startup time. Therefore, use the HPL only for large databases, for which the time savings in the actual loading or unloading of data makes the preparation time worthwhile. For other databases, use the data-migration tools that are discussed in this manual.

For information about using the HPL, refer to the [Guide to the High-Performance Loader](#). ♦

Database Server Environment Variables

Various *environment variables* affect the functionality of your Informix products. You can set environment variables that identify your terminal, specify the location of your software, and define other parameters. The environment variables discussed in this chapter are listed alphabetically beginning on [page A-3](#).

Some environment variables are required, and others are optional. For example, you must set—or accept the default setting for—certain UNIX or Windows NT environment variables.

This chapter describes how to use the environment variables that apply to Informix database servers and shows how to set them.

UNIX

NT

GLS

NLS

Types of Environment Variables

The environment variables discussed in this chapter fall into the following categories:

- **Informix environment variables**
Set these standard environment variables when you want to work with Informix products. Each product manual specifies the environment variables that you must set to use that product.
- **UNIX environment variables**
Informix products rely on the correct setting of certain standard UNIX system environment variables. The **PATH** and **TERM** environment variables must always be set. You might also have to set the **TERMCAP** or **TERMINFO** environment variable to use some products effectively. ♦
- **Windows NT environment variables**
Informix products rely on the correct setting of certain standard Windows NT system environment variables. The **PATH** environment variable must always be set. ♦
- **GLS environment variables**
The GLS environment variables that allow you to work in a nondefault locale are described in the [Guide to GLS Functionality](#). However, these variables are also included in the list of environment variables [on page A-3](#). ♦
- **NLS environment variables (pre-7.2 products only)**
You must set some or all of these X/Open standard environment variables to benefit from NLS. These environment variables might cause your product to behave differently than when their standard Informix counterparts are set. Refer to [“Native Language Support \(NLS\)” on page 9-8](#). ♦

List of Environment Variables

Figure A-1 contains an alphabetical list of the environment variables that you can set for an Informix database server. For instructions on setting these environment variables and detailed descriptions, see Chapter 4 of the [Informix Guide to SQL: Reference](#) for your server. The following table uses these abbreviations for the server names and Informix manuals:

- IUS (INFORMIX-Universal Server)
- ODS (INFORMIX-OnLine Dynamic Server)
- OWS (INFORMIX-OnLine Workgroup Server)
- SE (INFORMIX-SE)
- GLS ([Guide to GLS Functionality](#))
- REF ([Informix Guide to SQL: Reference](#))

GLS

The GLS environment variables are discussed in the [Guide to GLS Functionality](#). ♦

Figure A-1
Environment Variables for Informix Database Servers

Environment Variable	Restrictions	Reference
ARC_DEFAULT	IUS, ODS only (all platforms)	REF
ARC_KEYPAD	IUS, ODS only (all platforms)	REF
CLIENT_LOCALE	GLS only	GLS
CC8BITLEVEL	ESQL/C only	REF
COLLCHAR	NLS only	REF
DBANSIWARN		REF
DPAPICODE	NLS only	REF
DBBLOBBUF	IUS, ODS, OWS only (all platforms)	REF
DBCENTURY	SQL APIs only	REF
DBDATE		REF, GLS

(1 of 4)

List of Environment Variables

Environment Variable	Restrictions	Reference
DBDELIMITER		REF
DBEDIT		REF
DBFLTMASK	DB-Access only	REF
DBLANG		REF, GLS
DBMONEY		REF, GLS
DBNLS	NLS only	REF
DBONPLOAD	High-Performance Loader only	REF
DBPATH		REF
DBPRINT	UNIX only	REF
DBREMOTECMD	IUS, ODS, OWS only(UNIX)	REF
DBSPACETEMP	IUS, ODS, OWS only	REF
DBTEMP	SE only	REF
DBTIME	SQL APIs only	REF
DBUPSPACE		REF
DB_LOCALE	GLS only	GLS
DELIMIDENT		REF
ENVIGNORE		REF
ESQLMF	GLS only	GLS
FET_BUF_SIZE	SQL APIs and DB-Access only	REF
GLS8BITSYS	GLS only	GLS
GL_DATE	GLS only	GLS
GL_DATETIME	GLS only	GLS
INFORMIXC	ESQL/C only	REF
INFORMIXCONRETRY		REF

(2 of 4)

Environment Variable	Restrictions	Reference
INFORMIXCONTIME		REF
INFORMIXCONCSMCFG	IUS only	REF
INFORMIXDIR		REF
INFORMIXKEYTAB	IUS only	REF
INFORMIXOPCACHE	OnLine/Optical only	REF
INFORMIXSERVER		REF
INFORMIXSHMBASE	IUS, ODS, OWS only(UNIX)	REF
INFORMIXSQLHOSTS		REF
INFORMIXSTACKSIZE	IUS, ODS, OWS only (all platforms)	REF
INFORMIXTERM	DB-Access only	REF
INF_ROL_SEP	IUS, ODS only	REF
LANG	NLS only	REF
LC_COLLATE	NLS only	REF
LC_CTYPE	NLS only	REF
LC_MONETARY	NLS only	REF
LC_NUMERIC	NLS only	REF
LC_TIME	NLS only	REF
NODEFDAC		REF
ONCONFIG	IUS, ODS, OWS only (all platforms)	REF
OPTCOMPIND	IUS, ODS only (all platforms)	REF
PATH		REF
PDQPRIORITY	IUS, ODS only (all platforms)	REF
PLCONFIG	High-Performance Loader	REF
PSORT_DBTEMP	IUS, ODS only (all platforms)	REF

(3 of 4)

List of Environment Variables

Environment Variable	Restrictions	Reference
PSORT_NPROCS	IUS, ODS only (all platforms)	REF
SERVER_LOCALE	GLS only	GLS
SQLEXEC	ODS, SE only	REF
TERM	UNIX only	REF
TERMCAP	UNIX only	REF
TERMINFO	UNIX only	REF
THREADLIB	ESQL/C only	REF

(4 of 4)

Index

A

aaodir subdirectory 4-11, 4-15, 6-6, 8-18
 Access methods
 dropping indexes 3-25
 retrieving external data 3-6
 Administration mode 4-16, 6-7
 Administration tools
 configurations 6-4
 installing 4-26, 8-20, 8-29
 reinstalling 6-22
 reverting 4-44, 6-17
 uninstalling 4-46, 6-19
 upgrading 6-9
 Version 7.12 6-4, 6-17
 Version 7.22 6-4, 6-9
 adtcfg files 4-11, 4-15, 6-6, 8-18
 adtmasks files 4-15, 6-6, 8-18, 8-28
 AFF_NPROCS configuration
 parameter 3-5
 AFF_SPROC configuration
 parameter 3-5
 ALARMPROGRAM configuration
 parameter 4-37, 4-47, 6-20
 ALL keyword, with dbschema
 utility 10-40
 All Servers tree view 4-30, 6-14
 ALS8BITFSYS environment
 variable 9-10
 ALTER FRAGMENT
 statement 8-16
 ALTER TABLE statement 4-35
 ANSI compliance
 icon Intro-10
 level Intro-17
 ANSI database, moving 5-35

ANSI logging 5-36
 Answers OnLine 4-15, 6-6
 Applications, client
 ALS locales 9-11
 code-set conversion 9-13
 Communications Support
 Module 3-16
 GLS locales 9-7
 migrating 1-4
 NLS locales 9-9
 Archive-management utility 5-23
 ARC_CONFIG file
 saving 3-9
 updating 3-16
 ASCII file, produced by
 unloading 1-10, 9-5
 Asian Language Support (ALS)
 environment variables 9-10
 finding the locale 9-6
 migrating to Version 7.2 9-17
 reverting to ALS 9-22 to 9-24
 Version 7.2x support 9-11
 Audit configuration files 4-11, 8-18, 8-28
 Authentication policy 3-16, 3-28

B

Backing up
 logical logs 3-17
 OnLine Dynamic Server
 after migration 4-24, 5-37
 after reversion 4-50
 before migration 3-12, 4-18, 8-19
 Version 5.x 5-28, 5-43
 Version 7.22 4-30, 8-32

- OnLine Workgroup Server
 - before migration 8-22, 8-28
 - Version 7.12 6-9, 6-23
 - Version 7.22 6-15, 8-22
- Universal Server
 - after migration 3-18
 - before reversion 3-22
- Backup and Restore tool
 - Version 7.12 6-4, 6-9, 6-23
 - Version 7.22 4-30, 6-15, 8-22, 8-28
 - whole-system backup 8-22, 8-32
- bar utility (UNIX) 6-9
- bcheck utility, changed to
 - secheck 7-5
- Binary files, loading 1-11, 2-4, 10-59
- BladeManager
 - installing and registering
 - DataBlade modules 3-19
 - unregistering DataBlade
 - modules 3-23
- blademgr utility 3-23
- BLOB data type 3-24, 3-26
- Blobs
 - loading with dbload 10-25
 - moving with onload 2-6
- Blobspace
 - adding 8-7
 - backing up 3-4
 - disk space used 4-11
 - moving, with
 - onunload/onload 2-6
 - removing 8-13
- BOOLEAN data type 3-24
- boot90.sql and boot901.sql 3-20
- Bringing down
 - OnLine Dynamic Server 3-13, 4-20, 4-26, 5-28, 8-19
 - OnLine Workgroup Server 6-10, 8-29
- Bringing up
 - OnLine Dynamic Server 3-28, 4-23, 5-32, 5-43, 8-31
 - OnLine Workgroup Server 6-14, 6-23, 8-21
 - Universal Server 3-17
- B-tree index 3-25
- Buffer cache 5-7

- BUFFERS configuration
 - parameter 5-41
- BYTE data type 8-7

C

- Calculating memory requirements
 - for OnLine Dynamic Server 5-6
- Casts 3-25
- CC8BITLEVEL environment
 - variable 4-47, 6-21, 9-10, 9-12
- CDR_DSLOCKWAIT configuration
 - parameter 4-8
- CDR_EVALTHREADS
 - configuration parameter 4-8
- CDR_LOGBUFFERS configuration
 - parameter 4-8
- CDR_NIFMEMS configuration
 - parameter 4-8
- CDR_NIFQUEUES configuration
 - parameter 4-8
- CDR_NIFUSEHELP configuration
 - parameter 4-8
- CDR_QUEUEMEM configuration
 - parameter 4-8
- CHAR data type 9-23
- Character-position form of FILE
 - and INSERT statements 10-33
- CHECK TABLE statement 8-11
- Checking available space, Universal
 - Server 3-8
- Checking database integrity 3-18, 4-18, 5-43, 6-8, 8-18
- Choosing a migration
 - method 1-5 to 1-9
- CHUNKS configuration
 - parameter 5-41
- C-ISAM
 - converting to SE 7-6
 - migrating to OnLine Workgroup
 - Server or OnLine Dynamic
 - Server 1-8, 7-7
- Client applications
 - ALS locales 9-11
 - code-set conversion 9-13
 - Communications Support
 - Module 3-16
 - GLS locales 9-7
 - migrating 1-4
 - NLS locales 9-9
- CLIENT_LOCALE environment
 - variable 4-32, 6-16, 9-10, 9-12, 9-13
- CLOB data type 3-24, 3-26
- Code set
 - multibyte 9-10, 9-12
 - single-byte 9-8, 9-12
- Code-set conversion
 - HPL 2-9, 10-75
 - using 9-13
- COLLCHAR environment
 - variable 4-32, 4-48, 6-21, 9-8
- Collection data type 3-24
- Command Center
 - installing 8-20, 8-29
 - shutting down 6-10, 8-29
 - starting OnLine Dynamic
 - Server 4-30, 8-31
 - starting OnLine Workgroup
 - Server 6-14, 8-21
- Command file, dbload 10-27
- Command Line Utilities 6-8
- Command-line conventions
 - elements of Intro-11
 - example diagram Intro-13
 - how to read Intro-13
- Commands
 - export 5-30
 - setenv 5-30
 - See also* SQL statement.
 - See also* Utilities.
- Comment icons Intro-9
- COMMIT WORK statement 5-36
- Communications Support Module
 - client applications 3-16
 - configuring 3-16
 - removing 3-28
 - saving configuration 3-21
- Complete backup. *See* Whole-system backup.
- Compliance
 - icons Intro-10
 - with industry standards Intro-17
- conscm.cfg file
 - creating 3-16
 - deleting 3-28
 - saving 3-21

- Configuration file
 - OnLine Dynamic Server
 - customizing 4-28, 8-30
 - saving a copy 3-9, 4-15, 5-25, 8-18
 - setting up, example 5-30
 - OnLine Workgroup Server
 - customizing 6-13, 8-20
 - saving a copy 6-6, 8-28
 - Universal Server
 - customizing 3-15, 3-27
 - saving a copy 3-21
- Configuration parameter
 - AFF_NPROCS 3-5
 - AFF_SPROC 3-5
 - ALARMPROGRAM 4-37, 4-47, 6-20
 - BUFFERS 5-41
 - CHUNKS 5-41
 - continuous data replication 4-8, 6-20
 - DBSERVERALIASES 5-31
 - DBSERVERNAME 5-31
 - DBSPACES 5-41
 - DBSPACETEMP 5-15, 5-31
 - FILLFACTOR 5-37
 - HETERO_COMMIT 3-5, 4-47, 6-20
 - LOCKS 5-21
 - LOGFILES 5-41
 - LOGSMAX 5-41
 - MULTIPROCESSOR 3-5
 - NOAGE 3-5
 - NUMAIOVPS 3-5
 - NUMCPUVPS 3-5
 - ONLANGMAP 4-47, 6-20
 - OnLine Dynamic Server 5-21
 - OPTCOMPIND 4-10, 4-39
 - revised minimum values 5-21
 - ROOTOFFSET 3-15, 4-22, 4-39, 6-13, 8-20
 - ROOTPATH 3-15, 4-22, 6-13, 8-20
 - ROOTSIZE 3-15, 4-22, 6-13, 8-20
 - S BSPACENAME 3-5, 3-27
 - SHMADD 5-9, 5-22
 - SHMTOTAL 5-9, 5-22
 - SHMVIRT SIZE 5-22
 - SINGLE_CPU_VP 3-5
 - VPCLASS 3-5, 3-27

- Configurations, OnLine
 - Workgroup Server 6-4
- Configuring
 - open file descriptors 5-12
 - semaphore parameters 5-10
- config.arc file
 - saving 3-9
 - updating 3-16
- CREATE AUDIT statement 8-11
- CREATE DATABASE
 - statement 8-10, 8-15, 8-25, 8-35
- CREATE INDEX statement 5-19, 5-37
- CREATE PROCEDURE statement,
 - use of dbschema 10-41
- CREATE ROLE statement 8-16
- CREATE SYNONYM statement,
 - use of dbschema 10-41
- CREATE TABLE statement
 - loading data 8-10, 8-15, 8-25, 8-35
 - removing information 8-13, 8-23
 - use of dbschema 10-41
- CREATE VIEW statement, use of
 - dbschema 10-41
- Creating a dbload command
 - file 10-27
- CSM. *See* Communications Support Module.

D

- Data distributions and disk
 - utilization 5-13, 5-16
- Data type
 - BOOLEAN 3-24
 - BYTE 8-7
 - CHAR 9-23
 - CLOB and BLOB 3-24, 3-26
 - collection 3-24
 - DISTINCT 3-24
 - INT8 3-24
 - LVARCHAR 3-24
 - NCHAR 5-38, 9-16, 9-22, 9-24
 - NVARCHAR 5-38, 9-16, 9-22, 9-24
 - OPAQUE 3-24
 - ROW 3-24
 - SERIAL8 3-24

- TEXT 8-7
- VARCHAR 8-12, 9-23
- Database
 - displaying list of 6-8
 - migration. *See* Migrating a database.
 - ownership, set by onload 10-61
 - verifying integrity of 3-18, 4-17, 5-40, 6-14, 8-18
- Database Explorer 6-8
- Database locale. *See* Locale.
- Database server
 - administration mode 4-16, 6-7
 - platforms and versions 1-5, 1-8
 - reinstalling 4-48, 6-22
 - uninstalling 4-46, 6-19
 - upgrading 6-9
- Database server definitions 4-32, 6-16
- DataBlade modules
 - data types 3-24
 - deleting hidden tables 3-23
 - hidden tables 3-23
 - installing and registering 3-19
 - unregistering 3-23
- Data-replication secondary server
 - guidelines 5-23
- dbaccess utility
 - creating a database 8-10, 8-15
 - estimating size of user table indexes 5-17
 - updating statistics 5-14
 - using LOAD and UNLOAD 8-8, 8-14
- DBAPICODE environment
 - variable 4-7, 9-8, 9-13
- DBCENTURY environment
 - variable 4-47, 6-21
- DBCODESET environment
 - variable 9-10, 9-13
- DBC SOVERRIDE environment
 - variable 9-10
- DBDATE environment variable 8-8, 9-8, 9-10
- dbexport and dbimport utilities 5-5
- dbexport utility
 - c option 10-9
 - description of 10-7
 - destination options 10-10

- moving a database to
 - OnLine Dynamic Server 8-6, 8-33
 - OnLine Workgroup Server 8-6, 8-23
 - SE 8-12
- schema output 10-12
- ss option 8-6, 8-12, 10-9
- DBFLTMASK environment variable 4-47, 6-21
- DBFORMAT environment variable 9-10
- dbimport utility
 - c option 10-16
 - changing the locale 10-22
 - create options 10-19
 - database logging mode 10-17
 - delimited identifiers 10-16
 - description of 10-14
 - importing from another computer 2-8
 - input file location options 10-17
- Interrupt key 10-16
- l options 10-20
- moving a database to
 - OnLine Dynamic Server 8-8, 8-33
 - OnLine Workgroup Server 8-8, 8-24
 - SE 8-13
 - using with GLS 10-15
 - using with NLS 10-22
- DBINFXRC environment variable 9-10
- DBLANG environment variable 9-8, 9-10
- dbload utility
 - blobs, loading 10-25
 - compared to LOAD 10-27
 - creating a command file 10-27
 - description of 10-23
 - e option 10-26
 - FILE statement 10-27
 - i option 10-26
 - importing from another computer 2-8
- INSERT statement
 - compared to SQL INSERT statement 10-35
 - using 10-27, 10-30
- Interrupt key 10-25
- moving data to
 - OnLine Dynamic Server 8-10, 8-35, 8-37, 8-39
 - OnLine Workgroup Server 8-10, 8-25
 - SE 8-15
- number errors to allow 10-26
- r option 10-26
- speed, increasing 10-25
- syntax 10-23
- writing a command file
 - in character-position form 10-36
 - in delimiter form 10-31
- dblog utility, changed to selog 7-5
- DBMONEY environment variable 8-8, 9-8, 9-11
- DB-Monitor utility 5-28
- DBNLS environment variable 4-48, 6-21, 9-9, 9-13
- DBONPLOAD environment variable 4-47, 6-21
- DBPATH environment variable and sort files 5-15
- updating 8-16
- dbschema utility
 - create schema for a database 3-21, 10-41
 - description of 10-39
 - distribution information 10-51
 - moving data to
 - OnLine Dynamic Server 8-10, 8-37, 8-39
 - OnLine Workgroup Server 8-10
 - SE 8-15
- moving selected tables to
 - OnLine Dynamic Server 8-8, 8-37, 8-39
 - OnLine Workgroup Server 8-8
 - SE 8-14
- obtaining
 - privilege schema 10-46
 - synonym schema 10-45
- OnLine Dynamic Server
 - moving data 8-35
 - moving selected tables 8-34
- OnLine Workgroup Server
 - moving data 8-25
 - moving selected tables 8-24
 - owner conventions 10-42
 - saving database information 2-5
 - specifying a table, view, or procedure 10-49
 - specifying the role schema 10-48
 - ss option 10-43
- DBSERVERALIASES configuration parameter 5-31
- DBSERVERNAME configuration parameter 5-31
- DbSPACE
 - adding 8-7
 - backing up 3-4
 - defining disk space for
 - conversion 3-8, 5-13
 - data distributions 5-16
 - user table indexes 5-17
 - defining temporary
 - dbspaces 4-25, 5-15
 - free pages 5-17
 - moving tables to another
 - dbSPACE 2-7
 - removing 8-13
- DBSPACES configuration parameter 5-41
- DBSPACETEMP configuration parameter 5-15, 5-31
- DBSPACETEMP environment variable 5-31
- dbssodir subdirectory 8-18, 8-28
- DBTIME environment variable 9-9, 9-11
- DB_LOCALE environment variable 4-32, 6-16, 9-11, 9-12, 9-13
- Default locale Intro-6
- Delimiter form of FILE and INSERT statements 10-28
- Demonstration database Intro-7
- Diagnostics table, displayed by
 - dbschema utility 10-55
- Directory, installation 3-14, 4-20, 5-29
- Disk utilization
 - issues during conversion 5-13
 - root dbSPACE 5-13

- Display schema for a database 10-41
- DISTINCT data type 3-24
- Distribution information for tables 10-51
- dist_pages formula 5-16
- Documentation conventions
 - command-line Intro-11
 - icon Intro-9
 - sample-code Intro-14
 - typographical Intro-8
- Documentation notes 4-13, 6-5
- Documentation, types of
 - error message files Intro-16
 - machine notes Intro-16
 - on-line help Intro-15
 - on-line manuals Intro-15
 - printed manuals Intro-15
- DRDA database 2-8
- DROP AUDIT statement 8-11
- DROP INDEX SQL statement 5-19, 5-37
- DROP ROLE statement 8-16
- dropps.sql script 5-23

E

- EDA/SQL server. *See* Enterprise Gateway for EDA/SQL.
- Enterprise Gateway for EDA/SQL 2-8
- Enterprise Gateway Manager 2-8
- Enterprise Gateway with DRDA 2-8
- Enterprise Replication
 - configuration parameters 4-8
 - configuring 4-22
 - stopping 4-35
- Environment variable
 - ALS8BITFSYS 9-10
 - CC8BITLEVEL 4-47, 6-21, 9-10, 9-12
 - CLIENT_LOCALE 4-32, 6-16, 9-10, 9-12, 9-13
 - COLLCHAR 4-32, 4-48, 6-21, 9-8
 - DBAPICODE 4-7, 9-8, 9-13
 - DBCENTURY 4-47, 6-21
 - DBCODESET 9-10, 9-13
 - DBCVERIFY 9-10
 - DBDATE 8-8, 9-8, 9-10

- DBFLTMASK 4-47, 6-21
- DBFORMAT 9-10
- DBINFXRC 9-10
- DBLANG 9-8, 9-10
- DBMONEY 8-8, 9-8, 9-11
- DBNLS 4-48, 6-21, 9-9, 9-13
- DBONPLOAD 4-47, 6-21
- DBPATH 5-15, 8-16
- DBSPACETEMP 5-15, 5-31
- DBTIME 9-9, 9-11
- DB_LOCALE 4-32, 6-16, 9-11, 9-12, 9-13
 - definition of A-1
- ESQLMF 4-47, 6-21, 9-12
 - GLS8BITFSYS 4-48, 6-21, 9-12
 - GL_DATE 4-48, 6-21, 9-12
 - GL_DATETIME 4-48, 6-21, 9-12
- INFORMIXCONCSMCFG 3-4, 3-27
- INFORMIXDIR 5-30
- INFORMIXKEYTAB 3-4, 3-27
- INFORMIXSERVER 3-15, 5-21, 5-31, 8-8
- INFORMIXSQLHOSTS 3-15, 4-21
- LANG 4-48, 6-21, 9-9, 9-13
- LC_COLLATE 9-9
- LC_CTYPE 9-9
- LC_MONETARY 9-9
- LC_NUMERIC 9-9
- LC_TIME 9-9
- ONCONFIG 3-15, 4-13, 5-21, 5-30
- ONPLOAD 4-48, 6-21
- OPTCOMPIND 4-9, 4-25
- PATH 3-15, 5-30
- PLCONFIG 4-48, 6-21
- SERVER_LOCALE 4-48, 6-21, 9-12
- SQLEXEC 5-21
- TBCONFIG 4-13, 5-21
- THREADLIB 4-48, 6-21
 - types of A-2
- Environment variables
 - changes in OnLine Dynamic Server 6.0 5-21
 - list of all A-3 to A-6
 - reverting OnLine Dynamic Server 4-40, 4-47, 5-42
 - reverting OnLine Workgroup Server 6-21

- reverting Universal Server 3-27
- SE, required 7-4
- UNIX A-2
- Windows NT A-2
- en_us.819 locale name 9-17
- en_us.8859-1 locale Intro-6, 9-18, 9-23
- Error messages
 - DataBlade modules 3-23
 - files Intro-16
- ESQLMF environment
 - variable 4-47, 6-21, 9-12
- ESQL/C 9-13
- ESQL/COBOL 9-13
- Example
 - calculating free space 3-8
 - DataBlade module cleanup 3-23
 - onstat output 3-12
- export command 5-30
- Extended data types 3-19, 3-24
- Extents, checking 3-18, 4-18, 5-37, 6-8, 8-18
- External data, storage space 3-6, 3-26
- External spaces
 - creating 3-6
 - deleting 3-26
 - dropping virtual tables 3-26
- Extracting schema information 8-5
- Extspace. *See* External spaces.

F

- Fast recovery, initiating 3-10, 4-17, 6-7
- Feature icons Intro-9
- File descriptors 5-12
- FILE statement
 - character-position form 10-33
 - delimiter form 10-28
 - syntax for
 - character-position form 10-34
 - delimiter form 10-29
 - with dbload 10-27
- Files
 - documentation notes 4-13, 6-5
 - editing OnLine Dynamic Server configuration 5-30

- onconfig
 - OnLine Dynamic Server 3-9, 4-15, 5-25, 8-18
 - OnLine Workgroup Server 6-6, 8-28
 - Universal Server 3-15, 3-21
- onconfig.std 3-9, 4-15, 5-25
- readme 4-15, 6-6
- release notes 4-13, 5-3, 6-5
- secure auditing, growth of 5-20
- setting up onconfig 5-30
- sqlhosts 3-9, 3-15, 4-15, 5-31
- sqlhosts.demo 5-31
- sysmaster_pre60 5-23
- tbconfig 5-25
- tbconfig.std 5-25
- tctermcap
 - saving 3-9, 4-15, 5-25
 - used by ON-Archive 5-23
- termcap 3-9, 4-15, 5-25
- FILLFACTOR configuration parameter 5-37
- Formula
 - dist_pages 5-16
 - for growth_in_pages 5-14
 - free space, Universal Server 3-8
 - idx_growth_pages 5-17
- Fragmentation
 - adding 8-7
 - onload options 10-58
 - removing 8-13
- Free space required, Universal Server 3-8
- Functional indexes, Universal Server 3-25
- Functions, user-defined 3-23

G

- Gateway, importing from another computer 2-8
- Generic B-tree index 3-25
- glfiles utility 9-7
- Global Language Support (GLS) Intro-6
 - and dbimport utility 10-15
 - client locales 9-7
 - converting code-sets 9-13

- environment variables 9-12, A-2
- finding the locale 9-7
- migrating locales 4-32, 6-16
- NLS support 9-9
- onload and onunload 2-4
- reverting locales 4-36
- reverting to ALS locales 9-22
- reverting to NLS locales 4-46, 6-19, 9-20
- Universal Server 3-6
 - using onload and onunload 10-59
- GLS8BITFSYS environment variable 4-48, 6-21, 9-12
- GLS. *See* Global Language Support.
- GL_DATE environment variable 4-48, 6-21, 9-12
- GL_DATETIME environment variable 4-48, 6-21, 9-12
- GRANT FRAGMENT statement 8-16
- GRANT statement
 - role privileges 10-47
 - use of dbschema 10-41
- Group name 4-22, 4-29, 6-13
- growth_in_pages formula 5-14

H

- Hardware requirements
 - OnLine Dynamic Server 4-15
 - OnLine Workgroup Server 6-6
- HETERO_COMMIT configuration parameter 3-5, 4-47, 6-20
- High-Performance Loader (HPL)
 - converting code sets 10-75
 - loading ASCII or COBOL data 1-11, 2-9
 - OnLine Dynamic Server, supported 1-11, 8-32
 - OnLine Workgroup Server, not supported 1-11, 8-23
 - onpload utility 10-75
 - supported products 1-11, 2-9
 - using 1-11, 2-9, 10-75
- HPL. *See* High-Performance Loader.

I

- Icons
 - comment Intro-9
 - compliance Intro-10
 - feature Intro-9
 - platform Intro-9
 - product Intro-9
 - syntax diagram Intro-12
- Importing non-Informix data 1-4, 2-8
- Index
 - and dbload utility 10-26
 - B-tree 3-25
 - checking 3-18, 4-18, 5-43, 6-8, 8-18
 - rebuilding 3-17
 - R-tree 3-25
 - Universal Server 3-24
- Index space needed
 - idx_growth_pages formula 5-17
 - system catalog, disk utilization 5-14
- Industry standards, compliance with Intro-17
- Information Builders, Inc. *See* Gateway.
- Informix products. *See* Database server.
- informix user password 4-32, 6-16
- INFORMIXCONCSMCFG environment variable 3-4, 3-27
- INFORMIXDIR environment variable 5-30
- INFORMIX-Enterprise Gateway. *See* Gateway.
- INFORMIX-Gateway with DRDA. *See* Gateway.
- INFORMIXKEYTAB environment variable 3-4, 3-27
- INFORMIXSERVER environment variable
 - changing 8-8
 - OnLine Dynamic Server 5-21, 5-31
 - Universal Server 3-15
- INFORMIXSQLHOSTS environment variable
 - OnLine Dynamic Server 4-21
 - Universal Server 3-15

- Initial- and next-extent values 8-7, 8-13
- INSERT SQL statement 5-34
- INSERT statement
 - character-position form 10-33
 - delimiter form 10-28
 - syntax for
 - character-position form 10-34
 - delimiter form 10-29
 - with dbload 10-27
- inssnmp.exe utility 4-28, 6-11
- Installation directory 3-14, 4-20, 5-29
- Installing
 - administration tools 4-26
 - Command Center 8-20, 8-29
 - OnLine Dynamic Server 4-20, 4-27, 5-29, 8-29
 - OnLine Workgroup Server 6-10, 6-11, 8-20
 - Universal Server 3-14
- INT8 data type 3-24
- International database 9-3
- Interrupt key
 - with dbimport 10-16
 - with dbload 10-25
- Interruptions, preventing 5-39
- ISO 8859-1 code set Intro-6
- isql utility 5-17

L

- LANG environment variable 4-48, 6-21, 9-9, 9-13
- Language support. *See* Locale.
- LC_COLLATE environment variable 9-9
- LC_CTYPE environment variable 9-9
- LC_MONETARY environment variable 9-9
- LC_NUMERIC environment variable 9-9
- LC_TIME environment variable 9-9
- Level-0 backup
 - after moving data 2-5
 - OnLine Dynamic Server 3-12, 5-26

- OnLine Workgroup Server 6-9
- Universal Server 3-18
- LIST data type 3-24
- LOAD SQL statement
 - converting to non-default locales 5-38
 - moving selected tables to
 - OnLine Dynamic Server 8-10, 8-35
 - OnLine Workgroup Server 8-10, 8-25
 - SE 8-15
 - restriction 5-5
 - syntax 10-73
- Loading
 - ASCII files 1-10, 2-3
 - binary data 1-11, 2-4, 10-59
 - COBOL data 1-10, 1-11, 2-9
- Locale Intro-6
 - changing 9-5
 - compatibility problems 9-3
 - definition of 9-3
 - finding available locales 9-7
 - finding from systables 9-7, 9-14
 - finding information about 9-5
 - migrating from ALS to GLS 9-17
 - migrating from NLS to GLS 9-14
 - reverting from GLS to ALS 9-22
 - reverting from GLS to NLS 9-20
 - saving in system catalog 9-14
 - selecting 9-4
 - Version 7.2x products
 - using ALS 9-11
 - using GLS 9-12
 - using NLS 9-9
- Lock mode for tables 8-7, 8-13
- Locking, set by onload 10-61
- LOCKS configuration
 - parameter 4-10, 5-21
- LOGFILES configuration
 - parameter 5-41
- Logging, and
 - ANSI compliance 5-36
 - onload and onunload 2-5
- Logical logs
 - backing up 3-17, 4-23, 5-32
 - changing mode 8-13
 - logging status 8-23, 8-33

- minimum page total in OnLine Dynamic Server 5-22
- out of space 3-17
- restoring 3-18, 4-24, 5-37
- space required for Universal Server 3-8
- usage status 5-22
- LOGSMAX configuration
 - parameter 5-41
- Long transaction, risk when
 - converting indexes 5-36
- LTAPEDEV parameter
 - with onunload/onload 2-5
- LVARCHAR data type 3-24

M

- Machine notes Intro-16
- makeps.sql script 5-23
- Message segment, estimating size of 5-8
- Middle-Eastern language support 4-32, 6-16
- Migrating a database to different platforms 1-8
- GLS 9-14, 9-17
- OnLine Dynamic Server
 - using dbexport/dbimport 8-6, 8-33, 8-37, 8-39
 - using
 - UNLOAD/dbschema/LOAD 8-8, 8-34, 8-37, 8-39
- OnLine Workgroup Server
 - using dbexport/dbimport 8-6, 8-23
 - using
 - UNLOAD/dbschema/LOAD 8-8, 8-24
- overview 1-3
- same platforms 1-7
- SE
 - using dbexport/dbimport 8-12
 - using
 - UNLOAD/dbschema/LOAD 8-14

- Migrating between versions of
 - OnLine Dynamic Server 1-3,
 - 4-5 to 4-25, 5-3 to 5-38
 - SE 1-3, 7-4
- Migrating client applications 1-4
- Migration
 - onload utility 10-56
 - onunload utility 10-68
 - paths 1-5 to 1-9
- Migration procedure
 - between UNIX and Windows
 - NT 8-36 to 8-37, 8-38 to 8-39
 - from OnLine 4.1 or 5.0 to OnLine
 - Dynamic Server 7.2 5-24
 - from OnLine Dynamic Server to
 - OnLine Workgroup
 - Server 8-17 to 8-26
 - from OnLine Dynamic Server to
 - SE 8-12 to 8-16
 - from OnLine Dynamic Server to
 - Universal Server 3-6 to 3-19
 - from OnLine Workgroup Server
 - to OnLine Dynamic
 - Server 8-26 to 8-35
 - from OnLine Workgroup Server
 - to SE 8-12 to 8-16
 - from SE to OnLine Dynamic
 - Server 8-6 to 8-11
 - from SE to OnLine Workgroup
 - Server 8-6 to 8-11
- Mirroring 5-15
- Moving data
 - blobspaces 2-6
 - choosing a migration
 - method 1-5 to 1-9
 - choosing between dbload,
 - dbimport, and LOAD 1-13
 - C-ISAM, migrating 7-6
 - constraints 1-10
 - oncheck utility 2-6
 - overview 1-4
 - steps for using
 - onunload/onload 2-4
- Moving to another platform
 - migration paths 1-8
 - overview 1-4
- Multibyte character 9-10, 9-12

- MULTIPROCESSOR configuration
 - parameter 3-5
- MULTISET data type 3-24

N

- Native Language Support (NLS)
 - and GLS 5-38, 9-9
 - environment variables 9-8, A-2
 - finding the locale 9-6
 - migrating to Version 7.2 9-14
 - populating with dbimport 10-22
 - setting NLS locales 4-36, 4-46, 6-19
 - Version 7.2x support 9-9
- NCHAR data type 5-38, 9-16, 9-22,
 - 9-24
- NFILE, NFILES, NOFILE, NOFILES
 - OS parameters 5-12
- NLS. *See* Native Language Support.
- NOAGE configuration
 - parameter 3-5
- Non-Informix data, importing 1-4,
 - 2-8
- NUMAIOVPS configuration
 - parameter 3-5
- NUMCPUVPS configuration
 - parameter 3-5
- NVARCHAR data type 5-38, 9-16,
 - 9-22, 9-24

O

- ODS/NT. *See* OnLine Dynamic
 - Server.
- ODS/UNIX. *See* OnLine Dynamic
 - Server.
- Off-line mode, verifying 5-28, 8-19
- ON-Archive utility
 - and tctermcap file 5-23
 - and the sysmaster database 5-15
 - and Windows NT 4-18, 8-19
- backing up OnLine Dynamic
 - Server 3-12, 4-24, 5-37, 8-32
- backing up OnLine Workgroup
 - Server 6-9
- backing up Universal Server 3-18,
 - 3-22

- saving configuration files 3-9
 - updating configuration files 3-16
- ON-Bar utility, backing up
 - OnLine Dynamic Server 4-24, 8-32
 - OnLine Workgroup Server 6-9
 - Universal Server 3-18, 3-22
- oncheck utility
 - before moving data 2-6
 - changed from tbcheck 5-5
 - converting indexes 5-19, 5-35, 5-36
 - rebuilding table indexes 3-17
 - verifying database integrity 3-18,
 - 4-17, 5-40, 5-43, 6-8
- ONCONFIG environment variable
 - OnLine Dynamic Server 4-13,
 - 5-21, 5-30
 - Universal Server 3-15
- onconfig file
 - customizing 4-28, 6-13, 8-20
 - saving
 - OnLine Dynamic Server 3-9,
 - 4-15, 5-25, 8-18
 - OnLine Workgroup Server 6-6,
 - 8-28
 - Universal Server 3-21
 - updating 4-47, 6-20
- ONCONFIG file. *See* Configuration
 - file.
- ONCONFIG parameters. *See*
 - Configuration parameter.
- onconfig.std file 3-9, 4-15, 5-25
- oninit utility
 - OnLine Workgroup Server for
 - UNIX 6-14, 8-21
 - s option 3-10, 4-23
 - starting OnLine Dynamic
 - Server 4-17, 4-23, 5-32
 - starting Universal Server 3-17
- ONLANGMAP configuration
 - parameter 4-47, 6-20
- OnLine Dynamic Server
 - backing up
 - after migration 4-24, 5-37
 - after reversion 4-50
 - before migration 3-12, 4-18
 - Version 5.x 5-28, 5-43
 - Version 7.22 4-30, 8-32
 - Backup and Restore tool 8-32

- bringing down 3-13, 4-20, 4-26, 5-28, 8-19
- bringing up 3-28, 4-23, 5-32, 5-43, 8-31
- configuration issues 5-21
- configuring
 - open file descriptors 5-12
 - semaphore parameters 5-10
- converting user table
 - indexes 5-17 to 5-20, 5-33
- copying configuration files 3-9, 4-15, 5-25
- disk utilization 5-13
- environment variables A-3 to A-6
- environments supported 1-5
- estimating
 - memory requirements 5-6
 - size of system catalog
 - indexes 5-13
- HPL support 2-9
- initiating fast recovery 3-10, 4-17
- installing 4-20, 4-27, 5-29, 8-29
- migration
 - between UNIX and Windows
 - NT 1-8, 8-36 to 8-37
 - C-ISAM to ODS 7-7
 - summary 1-8
 - to OnLine Workgroup
 - Server 1-8, 8-17 to 8-23
 - to SE 1-8, 8-12 to 8-16
- moving data between
 - computers 2-3
- physical-memory
 - requirement 5-7
- preventing interruption of
 - users 5-39
- reinstalling 4-48
- release notes Intro-17, 4-13, 5-3
- reverting to
 - Version 4.x, 5.x 5-39 to 5-43, 9-23
 - Version 6.0 or later 4-33 to 4-43
- revise configuration parameter
 - values 5-21
- shutting down 4-26, 5-39, 8-19
- SQL syntax restrictions 8-16
- starting 3-28, 4-23, 5-32, 5-43
- uninstalling 4-46
- UNIX versions 4-33
- upgrading 4-26

- using data distributions 5-16
- using onload/onunload 2-3
- verifying
 - database integrity 4-17, 5-40
 - off-line mode 5-28, 8-19
- warning 4-46, 4-48, 8-27
- Windows NT versions 8-37
- On-line help Intro-15
- On-line manuals Intro-15
- OnLine Workgroup Server
 - backing up
 - before migration 8-22, 8-28
 - Version 7.12 6-9, 6-23
 - Version 7.22 6-15, 8-22
 - Backup and Restore tool 6-9, 8-22, 8-28
 - bringing down 6-10, 8-29
 - bringing up 6-14, 6-23, 8-21
 - Command Center 6-14
 - environment variables A-3 to A-6
 - environments supported 1-5
 - HPL support 2-9
 - initiating fast recovery 6-7
 - installing 6-9, 6-10, 6-11, 8-20
 - migration
 - between UNIX and Windows
 - NT 1-8, 8-38 to 8-39
 - C-ISAM to OWS 7-7
 - to OnLine Dynamic Server 1-8, 8-26 to 8-35
 - to SE 1-8, 8-12 to 8-16
 - moving data between
 - computers 2-3
 - PDQ support 8-23, 8-33
 - reinstalling 6-22
 - release notes Intro-17, 6-5
 - shutting down 6-10, 8-29
 - SQL syntax restrictions 8-16
 - uninstalling 6-19
 - upgrading 6-9
 - using onload/onunload 2-3
 - verifying database integrity 6-14, 8-18
 - versions 6-4
 - warning 6-19, 6-22
- onload and onunload
 - utilities 2-3 to 2-7, 5-5, 5-35

- onload utility
 - constraints on use 10-59
 - create options 10-58
 - logging status 10-60
 - moving a database 2-4
 - moving a table 2-6, 2-7
 - moving locales 10-59
 - moving to another dbspace 2-7
 - ownership and privileges 10-61
 - specifying tape parameters 10-57
 - steps for using 2-4
 - syntax 10-56
 - using between computers 1-11, 2-3
- onmode utility
 - and shared-memory
 - connections 5-10
 - b option 3-26, 4-45, 6-18, 10-63
 - changing mode to quiescent 5-35
 - ky option 3-10, 8-29
 - m option 3-28, 4-23, 5-33
 - quiescent to on-line mode 4-23, 5-33
 - reverting to OnLine 5.0 5-41, 10-64
 - reverting to OnLine 5.0 ALS 9-23
 - reverting to OnLine Dynamic
 - Server 6.0 10-65
 - reverting to OnLine Dynamic
 - Server 6.0 ALS 9-22
 - reverting to OnLine Dynamic
 - Server 7.12 4-46
 - reverting to OnLine Dynamic
 - Server 7.1x 4-36, 9-21
 - reverting to OnLine Workgroup
 - Server 7.12 6-19
 - reverting to Version 7.2 3-27, 10-66
 - s option 5-35
 - shutting down OnLine Dynamic
 - Server 3-10, 8-19
 - shutting down OnLine
 - Workgroup Server 8-29
 - sy option 3-10, 4-16, 6-7
 - syntax 10-63
 - treatment of systables 9-21
- ON-Monitor utility 3-13
- ONPLOAD environment
 - variable 4-48, 6-21
- onpload utility 10-75

- onsnmp utility, configuring 4-28, 6-11
- onspaces -d command 3-26
- onstat utility 3-12, 4-16
- ontape utility
 - backing up logical logs 4-23, 5-32
 - backing up OnLine Dynamic Server 3-12, 4-18, 4-24, 5-37, 8-19
 - backing up OnLine Workgroup Server 6-9
 - backing up Universal Server 3-18, 3-22
 - onunload/onload restriction 5-5
- onunload utility
 - constraints on use 10-59, 10-69
 - locking 10-71
 - logging mode 10-71
 - moving a database 2-4
 - moving a table 2-6, 2-7
 - moving locales 10-59
 - moving to another dbspace 2-7
 - ownership and privileges 10-70
 - steps for using 2-4
 - syntax 10-68
 - system catalog tables 10-71
 - use without tape parameters 10-68
 - using 1-11, 2-3
 - what is included with a database 10-70
 - table 10-71
- OPAQUE data type 3-24
- Operating system
 - reconfiguring 3-14, 4-20, 5-31
 - updates 5-12
- Operating-system parameter
 - NFILE 5-12
 - NFILES 5-12
 - NOFILE 5-12
 - NOFILES 5-12
 - SEMMNI 5-10
 - SEMMNS 5-11
 - SEMMSL 5-11
 - SHMMAX 5-9
 - SHMMNI 5-9
 - SHMSEG 5-10
 - SHMSIZE 5-9

- Operating-system resources
 - disk utilization 5-13
 - file descriptors 5-12
 - secure auditing log files 5-20
 - semaphores 5-10
 - shared memory 5-6 to 5-10
- Operators 3-25
- oper_deflt.arc file
 - saving 3-9
 - updating 3-16
- OPTCOMPIND configuration
 - parameter 4-10, 4-39
- OPTCOMPIND environment
 - variable 4-9, 4-25
- osahosts 4-27, 6-6, 6-11, 8-28
- Owner, in dbschema 10-42
- OWS/NT. *See* OnLine Workgroup Server.
- OWS/UNIX. *See* OnLine Workgroup Server.

P

- Parallel data query (PDQ) 8-23
- Parallel distributed queries, OnLine Dynamic Server 8-33
- PATH environment variable
 - OnLine Dynamic Server 5-30
 - Universal Server 3-15
- PDQ. *See* Parallel data query (PDQ).
- Performance comparison 3-18, 5-23
- Performing code-set conversion 9-13
- Physical memory requirement for OnLine Dynamic Server 5-7
- Planning
 - data migration overview 1-3
 - migrating to OnLine Workgroup Server 6-5
 - migrating to Universal Server 3-3
 - upgrading OnLine Dynamic Server 4-13, 5-6 to 5-23
- Platform icons Intro-9
- Platforms, moving data between
 - compatible computers 1-7, 2-3
 - operating systems 1-4, 1-8
- PLCONFIG environment
 - variable 4-48, 6-21
- Preparations for migrating to OnLine Dynamic Server 6.0 5-6
- Printed manuals Intro-15
- Privileges
 - access privileges for tables 10-71
 - required for onload 10-61
 - required for onunload 10-70
- Product icons Intro-9
- Pseudotables, in sysmaster database 5-15

R

- Readme file 4-15, 6-6
- read_ods.txt file 4-15
- Rebuild system catalog
 - indexes 5-33
- Recompiling client applications 1-4
- Reconfiguring the operating system 3-14, 4-20, 5-31
- RECOVER TABLE statement 8-11
- Registering DataBlade modules 3-19
- Registry keys
 - SNMP master agent 4-28, 6-11
 - sqlhosts 4-27, 6-11
- Reinstalling
 - OnLine Dynamic Server 4-48
 - OnLine Workgroup Server 6-22
- Release notes
 - OnLine Dynamic Server 4-13, 5-3
 - OnLine Workgroup Server 6-5
 - Universal Server 3-3
- Removing administration tools 4-44
- RENAME DATABASE
 - statement 8-16
- REPAIR TABLE statement 8-11
- Replication Manager 4-22, 4-29, 6-13
- Requirements, system 4-15, 6-6
- Reserve pages, checking 3-18, 4-18, 5-43, 6-8, 8-18
- Resident segment, estimating size of 5-8
- Restoring
 - OnLine Dynamic Server data 4-50
 - OnLine Workgroup Server data 6-9, 6-23

- Reverting
 - administration tools 4-44, 6-17
 - Enterprise Replication 4-35
 - GLS locales to ALS 9-22
 - GLS locales to NLS 9-20
 - modifying configuration parameters 4-47, 6-20
 - OnLine Dynamic Server to
 - 6.0 or later 4-33 to 4-43, 10-63
 - pre-6.0 5-39 to 5-43, 10-63
 - reinstalling the product 4-48
 - resetting environment variables 4-47
 - Universal Server 3-20 to 3-28
 - Workgroup Replication 4-45, 6-18
- REVOKE FRAGMENT statement 8-16
- ROLLBACK WORK statement 5-36
- ROLLFORWARD DATABASE statement 8-11
- Root dbspace
 - disk utilization 5-13
 - free pages 5-15
- ROOTOFFSET configuration
 - parameter 3-15, 4-22, 4-39, 6-13, 8-20
- ROOTPATH configuration
 - parameter 3-15, 4-22, 6-13, 8-20
- ROOTSIZE configuration
 - parameter 3-15, 4-22, 6-13, 8-20
- Routines, user-defined 3-23
- ROW data type 3-24
- RSAM shared-memory segment 5-7
- R-tree indexes 3-25

S

- Sample-code conventions Intro-14
- SBSPECNAME configuration
 - parameter 3-5, 3-27
- Sbspaces
 - about 3-6
 - deleting 3-26
- SE
 - converting C-ISAM files 7-6
 - converting unsupported data types 8-12
 - environment variables A-3 to A-6

- migrating to
 - different versions 7-4
 - OnLine Dynamic Server 1-9, 8-6 to 8-11
 - OnLine Workgroup Server 1-9, 8-6 to 8-11
- onload and onunload 2-4
- reverting to an earlier version 7-6
- SQL syntax restrictions 8-11
- steps for migration 7-5
- version levels 1-5
- Secure auditing, growth of log files 5-20
- Segment
 - message 5-7
 - resident 5-7
 - virtual 5-7
- SELECT statement 5-17, 5-18, 5-20, 9-6 to 9-7
- Semaphores 5-10
- SEMMNI OS parameter 5-10
- SEMMNS OS parameter 5-11
- SEMMSL OS parameter 5-11
- SERIAL data type, treatment by dbschema 10-41
- Serial number and key 4-27, 6-10
- SERIAL8 data type 3-24
- Server menu 4-30, 6-14
- Server, database. *See* Database server.
- SERVER_LOCALE environment variable 4-48, 6-21, 9-12
- Services file 8-20, 8-30
- Services, Windows NT 8-19
- SET CONSTRAINTS statement 8-16
- SET data type 3-24
- SET DATASKIP statement 8-16
- SET EXPLAIN statement 4-25
- SET ISOLATION statement 8-16
- SET LOG statement 8-16
- SET PDQPRIORITY statement 8-16
- SET ROLE statement 8-16
- SET SESSION AUTHORIZATION statement 8-16
- setenv command 5-30
- Setnet32 tool 4-28, 6-13, 6-21
- Setup tool 4-32, 6-16
- Shared server machine 4-32, 6-16

- Shared-memory communication interface 5-10
- Shared-memory segments
 - added at run time 5-9
 - estimating size of 5-6 to 5-10
 - in the operating system 5-9
- SHMADD configuration
 - parameter 5-9, 5-22
- SHMMAX OS parameter 5-9
- SHMMNI OS parameter 5-9
- SHMSEG OS parameter 5-10
- SHMSIZE OS parameter 5-9
- SHMTOTAL configuration
 - parameter 5-9, 5-22
- SHMVIRTSIZE configuration
 - parameter 5-8, 5-22
- Shutting down
 - Enterprise Replication 4-35
 - OnLine Dynamic Server 4-26, 5-39, 8-19
 - OnLine Workgroup Server 6-10, 8-29
 - Workgroup Replication 4-45, 6-18
- Simple large objects
 - loading with dbload 10-25
 - moving with onload 2-6
- Single-byte character 9-8, 9-12
- SINGLE_CPU_VP configuration
 - parameter 3-5
- Smart large objects
 - creating storage space 3-6
 - data types 3-24
 - deleting storage space 3-26
 - removing 3-24
- smi_load utility 10-66
- smi_unld utility 10-65
- SNMP, configuring 4-28, 6-11
- Software dependencies Intro-6
- Solaris server agent 6-11
- Sort files and DBSPACETEMP 5-15
- Space
 - additional pages for system catalog indexes 5-14
 - for sysmaster database 3-8, 5-15
- Space requirements
 - OnLine Dynamic Server 4-15
 - OnLine Workgroup Server 6-6

SQL statement

- ALTER FRAGMENT 8-16
- CHECK TABLE 8-11
- COMMIT WORK 5-36
- CREATE AUDIT 8-11
- CREATE DATABASE 8-10, 8-15, 8-25, 8-35
- CREATE INDEX 5-19, 5-37
- CREATE ROLE 8-16
- CREATE TABLE
 - loading data 8-10, 8-15, 8-25, 8-35
 - removing information 8-13, 8-23
- DROP AUDIT 8-11
- DROP INDEX 5-19, 5-37
- DROP ROLE 8-16
- FILE 10-27
- GRANT FRAGMENT 8-16
- INSERT 5-34, 10-27
- LOAD 5-5, 5-38
- RECOVER TABLE 8-11
- remove unsupported syntax 4-34, 6-17, 8-9, 8-23, 8-33
- RENAME DATABASE 8-16
- REPAIR TABLE 8-11
- REVOKE FRAGMENT 8-16
- ROLLBACK WORK 5-36
- ROLLFORWARD
 - DATABASE 8-11
- SELECT 5-17, 5-18, 5-20, 9-6 to 9-7
- SET DATASKIP 8-16
- SET EXPLAIN 4-25
- SET LOG 8-16
- SET PDQPRIORITY 8-16
- SET ROLE 8-16
- SET SESSION
 - AUTHORIZATION 8-16
- START DATABASE 8-11
- syntax recognized by
 - OnLine Dynamic Server 8-16
 - OnLine Workgroup Server 8-16
- SE 8-11
- UNLOAD 5-5, 5-38
- UPDATE STATISTICS
 - data distributions 5-16, 10-51
 - queries 4-24, 8-22, 8-32
 - system catalog indexes 5-14
 - user table indexes 5-19, 5-36

SQLEXEC environment

- variable 5-21
- sqlhosts file, UNIX
 - changing name or path 3-15
 - csn option 3-16, 3-28
 - modify, example 5-31
 - pre-6.0 OnLine Dynamic Server 7-4
 - save a copy when migrating 3-9, 4-15, 5-25, 8-28
- sqlhosts, Windows NT
 - database server definitions 8-20, 8-30
 - registry 4-27, 6-11
- sqlhosts.demo file 5-31
- sqlturbo processes 5-22
- srv_agent2 8-30
- START DATABASE statement 8-11
- Starting
 - OnLine Dynamic Server 3-28, 4-23, 5-32, 5-43, 8-31
 - OnLine Workgroup Server 6-14, 6-23, 8-21
 - Universal Server 3-17
- Stored procedures, Universal Server 3-25
- stores7 database Intro-7
- Swap space, size of 5-7
- Syntax conventions
 - icons used in Intro-12
- sysbldiprotected table 3-23
- sysbldirequired table 3-23
- sysbldobjdepends table 3-23
- sysbldobjects table 3-23
- sysbldregistered table 3-23
- syserrors catalog, UGEN errors 3-23
- sysmaster database
 - and disk utilization 5-13
 - and logical logs 3-17, 5-22
 - automatic creation of 5-32
 - pre-existing 5-23
 - space required to build 3-8, 4-15, 5-15
 - unload before reversion 10-65
- sysmaster_pre60 file 5-23
- systables, finding the locale 9-7, 9-14

System catalog indexes

- disk utilization 5-14
- estimating pages needed 5-13
- rebuild 5-33

System catalogs

- boot scripts, Universal Server 3-20
- checking tables 3-18, 4-18, 5-43, 6-8, 8-18
- reverting Universal Server 3-26
- size of, Universal Server 3-17
- storing database locale 9-14

System resources, additional required for 6.0 5-6

T

Tables

- estimate additional index
 - pages 5-18
- finding location of 5-18
- in separate dbspaces, disk utilization 5-18
- moving to another dbspace 5-34

TAPEDEV parameter

- with onunload/onload 2-5

tbcheck utility

- replaced by oncheck 4-13, 5-5
- verifying the integrity of data 5-27

TBCONFIG environment

- variable 4-13, 5-21

tbconfig file 5-25

tbconfig.std file 5-25

tbinit utility 5-26, 5-42

tbload and tbunload utilities 5-5

tbmode utility 5-26, 5-28

tbmonitor utility 5-28

tbstat utility

- option 5-27
- d option 5-15, 5-17, 5-25
- l option 5-22

tbtape utility 5-5

tctermcap file

- saving 3-9, 4-15, 5-25
- used by ON-Archive 5-23

Temporary files and tables 5-15

termcap file in

- SINFORMIXDIR/etc 3-9, 4-15, 5-25

TEXT data type 8-7
Text file, description of 9-5
THREADLIB environment
variable 4-48, 6-21
Tools, migration 1-9

U

Uninstall tool 4-46, 6-19
Uninstalling
administration tools 4-46, 6-19
OnLine Dynamic Server 4-46
OnLine Workgroup Server 6-19
Universal Server
authentication policies 3-16, 3-28
backing up
after migration 3-18
before reversion 3-22
bringing up 3-17
casts 3-25
DataBlade modules,
installing 3-19
dropping indexes 3-24, 3-25
environments supported 1-5
extended data types 3-24
GLS support 3-6
guidelines 3-19
installing 3-14
migration planning 3-3
migration procedure 3-6 to 3-19
operators 3-25
release notes 3-3
reversion utility 3-26
reverting 3-20 to 3-28
space requirements 3-8, 3-17
starting 3-17
stored procedures 3-25
system catalogs 3-17
user-defined data types 3-24
user-defined routines 3-23
using onload/onunload 2-4
UNIX
environment variables A-2
migration paths 1-8, 1-9
OnLine Dynamic Server
versions 1-5, 4-33
OnLine Workgroup Server
versions 1-5

starting OnLine Workgroup
Server 6-14, 8-21
Universal Server versions 1-5
UNLOAD SQL statement
moving data to
OnLine Dynamic Server 8-10,
8-35, 8-37
OnLine Workgroup Server 8-10,
8-25, 8-39
SE 8-15
moving selected tables to
OnLine Dynamic Server 8-9,
8-34
OnLine Workgroup Server 8-9,
8-24
SE 8-14
restriction 5-5
syntax 10-74
using nondefault locales 5-38
Unregistering DataBlade
modules 3-23
UPDATE STATISTICS statement
data distributions 5-16, 10-51
queries 4-24, 8-22
system catalog indexes 5-14
user table indexes 5-19, 5-36
Updating ON-Archive
configuration 3-16
Upgrade screen 4-27, 6-10
Upgrading
from OnLine 5.0 5-3 to 5-38
from OnLine Dynamic Server 6.0
or later 4-5 to 4-25
scenarios, OnLine Workgroup
Server 6-4
to Universal Server 3-6 to 3-19
User table indexes
and disk utilization 5-13
converting in OnLine Dynamic
Server 5-33
preparing for conversion to
6.0 5-18
risk of long transaction when
converting 5-36
User-defined index, Universal
Server 3-25
User-defined routines and
functions 3-23
USERS configuration parameter 5-8

Utilities

changes, OnLine Dynamic Server
6.0 5-5
dbaccess 5-14, 5-17
dbexport 5-5, 10-7
dbimport 5-5, 10-14
dbload 10-23
DB-Monitor 5-28
dbschema 10-39
dropps.sql script 5-23
glfiles 9-7
isql 5-17
makeps.sql script 5-23
ON-Archive. *See* ON-Archive
utility.
ON-Bar. *See* ON-Bar utility.
oncheck. *See* oncheck utility.
oninit. *See* oninit utility.
onload and onunload 2-3 to 2-7,
5-5, 5-35
onmode. *See* onmode utility.
ON-Monitor 3-13
onpload 10-75
onspaces 3-26
onstat 3-12
ontape. *See* ontape utility.
smi_load 10-66
smi_unld 10-65
tbcheck 5-5, 5-27
tbinit 5-26, 5-42
tbload and tbunload 5-5
tbmode 5-26, 5-28
tbmonitor 5-28
tbstat. *See* tbstat utility.
tbtape 5-5

V

VARCHAR data type 8-12, 9-23
Verifying
database integrity 3-18, 5-37, 6-8
off-line mode 8-19
Versions, Informix database
servers 1-5, 1-8
View, display description with
dbschema 10-41
Virtual processors 5-10

- Virtual segment
 - estimating the size of 5-8
 - use of 5-7, 5-22
- Virtual tables 3-26
- VPCLASS configuration
 - parameter 3-5, 3-27

W

- Whole-system backup 4-30, 6-15
- Windows NT
 - environment variables A-2
 - migration paths 1-8
 - OnLine Dynamic Server
 - versions 1-5, 8-37
 - OnLine Workgroup Server
 - versions 1-5
- Workgroup Replication
 - configuration parameters 4-8, 6-20
 - configuring 4-28, 4-29, 6-11, 6-13
 - stopping 4-45, 6-18
- Writing a dbload command file
 - in character-position form 10-36
 - in delimiter form 10-31

Symbols

- \$ARC_CONFIG. *See* ARC_CONFIG file.